



APRENDA AZURE EN UN MES DE ALMUERZOS

Segunda edición
Iain Foulds

} *

|||||||

#



Obtenga ayuda
con su proyecto.

Hable con un
especialista
en ventas >

startHere(Azure);

Profundice en las 21 lecciones de Azure

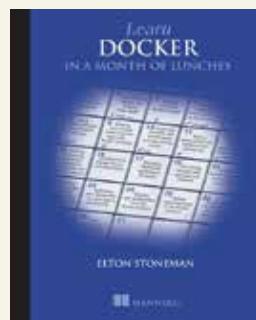
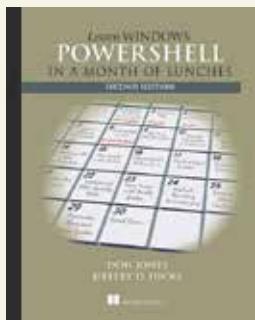
Obtenga una base sólida en Azure con las lecciones de este eBook. Regístrese para obtener una cuenta gratuita de Azure y utilice su crédito de USD 200 para completar los ejercicios. Continúe con su cuenta y obtenga 12 meses de servicios populares gratuitos y más de 25 servicios siempre gratuitos.



Comience gratis

¡AHORRE 40 % EN LIBROS Y VIDEOS DE MANNING!

Manning publica libros y videos de alta calidad para profesionales de la tecnología como usted. Utilice este **código de descuento especial para ahorrar un 40 % en todos los eBooks, pBooks, MEAP y cursos de liveVideo en manning.com, incluidos estos títulos seleccionados. Solo tiene que escribir azuremsft2** en el cuadro de Código promocional en el momento de realizar el pago.



Aprenda Windows PowerShell en un mes de almuerzos

por Don Jones y Jeffery Hicks

Diciembre de 2016, 384 páginas

Aprenda Docker en un mes de almuerzos

Elton Stoneman

Verano de 2020, 530 páginas

Más libros para un mes de almuerzos

[Aprenda Windows PowerShell en un mes de almuerzos, tercera edición](#)

[Aprenda Docker en un mes de almuerzos](#)

[Aprenda dbatools en un mes de almuerzos](#)

[Aprenda PowerShell en un mes de almuerzos, Linux y macOS Edition](#)

[Aprenda a crear scripts con PowerShell en un mes de almuerzos](#)

[Aprenda Linux en un mes de almuerzos](#)

[Aprenda Amazon Web Services en un mes de almuerzos](#)

[Aprenda administración de redes de Cisco en un mes de almuerzos](#)

Libros para desarrolladores de Microsoft y profesionales de TI

[Ingeniería de Azure Data](#)

[Principios, prácticas y patrones de inserción de dependencias](#)

[Microservicios en .NET Core](#)

[.NET Core en acción](#)

[Seguridad de los microservicios en acción](#)

[Simultaneidad en .NET](#)

[Aplicaciones reactivas con Akka.NET](#)

[ASP.NET Core en acción](#)

[Entity Framework Core en acción](#)

[C# en profundidad, cuarta edición](#)

[Programación funcional en C#](#)

[Kubernetes en acción](#)

[Knative en acción](#)

[Microservicios de arranque](#)

[con Docker, Kubernetes y Terraform](#)

[Core Kubernetes](#)

[GitOps y Kubernetes](#)

[Docker en acción, segunda edición](#)

[Docker en la práctica, segunda edición](#)

[Docker en movimiento](#)

[OpenShift en acción](#)

[Patrones nativos de la nube](#)

Lea libros de Manning GRATIS en liveBook

La plataforma liveBook de Manning ofrece una experiencia de lectura en línea cómoda y flexible. Obtiene **GRATIS acceso completo durante cinco minutos al día** a cada libro de Manning. En liveBook, puede

- Hacer preguntas, compartir código y ejemplos, e interactuar con otros lectores en el foro de liveBook.
- Buscar texto completo en todos los libros de Manning, incluso libros que no posea.
- Registrarse para obtener GRATIS una cuenta de liveBook en livebook.manning.com

Puede usar sus cinco minutos GRATIS como quiera: inicie y detenga el temporizador, salte entre libros y pruebe los ejercicios interactivos. Solo tiene que iniciar sesión y **explorar sin riesgos**.

Elogio a la primera edición

De la primera edición de *Aprenda Azure en un mes de almuerzos* de Iain Foulds:

"Un libro increíble y repleto de información para aprender conceptos básicos y avanzados de Azure en un mes".

—Sushil Sharma, Galvanize

"Microsoft Azure se está convirtiendo rápidamente en un líder en el espacio de la nube pública. Al seguir los ejercicios de este libro, se pondrá rápidamente al día con esta tecnología".

—Michael Bright, asesor para los desarrolladores, consultor independiente

"Excelente introducción a Azure con muchos ejemplos prácticos. Incluye una gran gama de temas actuales".

—Sven Stumpf, ING-DiBa AG

"Azure es como un océano. Este libro lo mantiene a flote al brindarle la mejor manera de aprender en forma de almuerzos ricos en práctica y ejemplos".

—Roman Levchenko, Microsoft MVP

"Todo lo que un desarrollador ocupado necesita para empezar a trabajar en Azure".

—Rob Loranger, desarrollador independiente

"Una excelente manera de comprender la amplitud de las ofertas de Azure siguiendo un enfoque breve centrado en las actividades".

—Dave Corun, Avanade

"El libro más completo sobre Azure que he encontrado para empezar a desarrollar mis proyectos académicos".

—Marco Giuseppe Salafia, estudiante de doctorado, Università degli Studi di Catania

"Este es el libro de referencia para la plataforma Azure. Está bien organizado, es exhaustivo y completo. Empezando por lo básico, guía al lector a través de la creación de configuraciones cada vez más complejas con la plataforma Azure para brindar escalabilidad, alto rendimiento y redundancia para las aplicaciones y los servicios hospedados. Este libro servirá tanto de tutorial para el principiante como de referencia para el usuario más experimentado".

—Robert Walsh, Excalibur Solutions

*Aprenda Azure en un mes
de almuerzos*

SEGUNDA EDICIÓN

IAIN FOULDS



MANNING
SHELTER ISLAND

Para obtener información y realizar pedidos en línea de este y otros libros de Manning, visite www.manning.com. El editor ofrece descuentos en este libro cuando se realizan pedidos en grandes cantidades. Para obtener más información, póngase en contacto con:

Departamento de ventas especiales
Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964
Correo electrónico: orders@manning.com

©2020 por Manning Publications Co. Todos los derechos reservados.

Ninguna parte de esta publicación se podrá reproducir, almacenar en un sistema de recuperación ni transmitir de ninguna forma ni por ningún medio electrónico, mecánico, fotocopiado ni de ningún tipo sin el consentimiento previo por escrito del editor.

Muchas de las designaciones que usan los fabricantes y vendedores para distinguir sus productos se consideran marcas registradas. Cuando dichas designaciones aparecen en el libro, y Manning Publications estaba al tanto de que se trataban de marcas registradas, las designaciones están con mayúsculas iniciales o completamente en mayúsculas.

- ⊗ Debido a que se reconoce la importancia de preservar lo que se ha escrito, es política de Manning imprimir los libros en papel sin ácido, para lo que realizamos nuestros mayores esfuerzos. También reconocemos nuestra responsabilidad de conservar los recursos de nuestro planeta, por lo cual los libros de Manning se imprimen en papel que es, al menos, un 15 % reciclado y procesado sin el uso de cloro elemental.



Manning Publications Co.
20 Baldwin Road
PO Box 761
Shelter Island, NY 11964

Editor de adquisiciones:	Mike Stephens
Editor de desarrollo:	Frances Lefkowitz
Editor de desarrollo técnico:	Karsten Strøbaek
Editor de revisión:	Aleksandar Dragosavljevic
Editor de producción:	Anthony Calcara
Editor de gráficos:	Jennifer Houle
Correctora de estilo:	Kathy Simpson
Revisor:	Katie Tennant
Corrector técnico:	Karsten Strøbaek
Cajista:	Marija Tudor
Diseñador de portada:	Leslie Haimes

ISBN 9781617297625
Publicado en los Estados Unidos de América.

*A los pilares de mi vida:
Abigail, Bethany y Charlotte*

contenido

<i>prefacio</i>	xv
<i>agradecimientos</i>	xvi
<i>acerca de este libro</i>	xvii
<i>acerca del autor</i>	xxi

PARTE 1 SERVICIOS PRINCIPALES DE AZURE1

1 Antes de comenzar 3

1.1	¿Es este el libro que busca?	3
1.2	Cómo usar este libro	4
	<i>Los capítulos principales</i>	4
	<i>Pruébelo ahora</i>	5
	<i>Laboratorios prácticos</i>	5
	<i>Código fuente y materiales suplementarios</i>	5
1.3	Creación de su entorno de laboratorio	5
	<i>Creación de una cuenta gratuita de Azure</i>	5
	<i>Ejercicio de laboratorio adicional: creación de una cuenta gratuita de GitHub</i>	7
1.4	Un poco de ayuda	7
1.5	Comprensión de la plataforma Azure	8
	<i>Virtualización en Azure</i>	10
	<i>Herramientas de administración</i>	11

2 Creación de una máquina virtual 14

2.1	Conceptos básicos de la configuración de la máquina virtual	15
	<i>Regiones y opciones de disponibilidad</i>	15
	<i>Imágenes de VM</i>	16
	<i>Tamaños de VM</i>	17
	<i>Almacenamiento de Azure</i>	18
	<i>Redes virtuales</i>	19

- 2.2 Creación de un par de claves SSH para la autenticación 20
- 2.3 Creación de una VM desde su navegador web 22
- 2.4 Conexión a la VM e instalación del servidor web 24
 - Conexión a la VM con SSH* 24 ▪ *Instalación del servidor web* 26
- 2.5 Cómo permitir que el tráfico web llegue a la VM 27
 - Creación de una regla para permitir el tráfico web* 28 ▪ *Visualización del servidor web en acción* 28
- 2.6 Laboratorio: creación de VM de Windows 29
- 2.7 Limpieza de recursos 30
- 2.8 Houston, tenemos un problema 31

3 Azure Web Apps 33

- 3.1 Descripción general y conceptos de Azure Web Apps 34
 - Lenguajes y entornos compatibles* 34 ▪ *Representación de versiones diferentes mediante ranuras de implementación* 35 ▪ *Planes de App Services* 35
- 3.2 Creación de una aplicación web 37
 - Creación de una aplicación web básica* 37 ▪ *Implementación de un sitio HTML de ejemplo* 39
- 3.3 Visualización de registros de diagnóstico 42
- 3.4 Laboratorio: creación y utilización de una ranura de implementación 44

4 Introducción a Azure Storage 47

- 4.1 Discos administrados 47
 - Discos de SO* 48 ▪ *Discos temporales y discos de datos* 49
Opciones de almacenamiento en caché de disco 50
- 4.2 Cómo agregar discos a una VM 50
- 4.3 Azure Storage 52
 - Almacenamiento de tablas* 53 ▪ *Almacenamiento de colas* 55 ▪ *Disponibilidad y redundancia de almacenamiento* 56
- 4.4 Laboratorio: exploración de Azure Storage 57
 - Centrado en la VM* 57 ▪ *Centrado en el desarrollador* 57

5 Conceptos básicos de Redes de Azure 58

- 5.1 Componentes de las redes virtuales 58
 - Redes y subredes virtuales* 59 ▪ *Tarjetas de interfaz de red virtuales* 61 ▪ *Dirección IP pública y resolución DNS* 62

- 5.2 Protección y control de tráfico con grupos de seguridad de red 64
 - Creación de un grupo de seguridad de red* 64 ▪ *Asociación de un grupo de seguridad de red con una subred* 66 ▪ *Creación de reglas de filtrado de grupo de seguridad de red* 67
- 5.3 Creación de una aplicación web de ejemplo con tráfico seguro 68
 - Creación de conexiones de red con acceso remoto* 68 ▪ *Creación de VM* 69 ▪ *Uso de agentes de SSH para conectarse a VM* 70
- 5.4 Laboratorio: instalación y prueba del servidor web LAMP 72

PARTE 2 ALTA DISPONIBILIDAD Y ESCALABILIDAD.....73

6 Azure Resource Manager 75

- 6.1 El enfoque de Azure Resource Manager 75
 - Diseño alrededor del ciclo de vida de las aplicaciones* 76 ▪ *Protección y control de recursos* 78 ▪ *Protección de recursos con bloqueos* 79 ▪ *Administración y agrupación de recursos con etiquetas* 80
- 6.2 Plantillas de Azure Resource Manager 81
 - Creación y uso de plantillas* 82 ▪ *Creación de varios tipos de recursos* 84 ▪ *Herramientas para compilar sus propias plantillas* 85 ▪ *Almacenamiento y uso de plantillas* 87
- 6.3 Laboratorio: implementación de recursos de Azure desde una plantilla 87

7 Alta disponibilidad y redundancia 90

- 7.1 La necesidad de redundancia 90
- 7.2 Redundancia de infraestructura con zonas de disponibilidad 92
 - Creación de recursos de red en una zona de disponibilidad* 94
 - Creación de VM en una zona de disponibilidad* 95
- 7.3 Redundancia de VM con conjuntos de disponibilidad 96
 - Dominios de error* 96 ▪ *Dominios de actualización* 97 ▪ *Distribución de las VM en los conjuntos de disponibilidad* 98 ▪ *Visualización de distribución de las VM en conjuntos de disponibilidad* 101
- 7.4 Laboratorio: implementación de VM altamente disponibles desde una plantilla 102

8 Aplicaciones de equilibrio de carga 106

- 8.1 Componentes del equilibrador de carga de Azure 106

Creación de un grupo IP de front-end 108 ▪ Creación y configuración de sondeos de estado 110 ▪ Definición de la distribución de tráfico con reglas del equilibrador de carga 112 ▪ Enrutamiento de tráfico directo con reglas de traducción de direcciones de red 114 ▪ Asignación de grupos de VM a grupos de back-end 116

- 8.2 Creación y configuración de VM con el equilibrador de carga 119
- 8.3 Laboratorio: visualización de plantillas de implementaciones existentes 122

9 Aplicaciones que escalan 124

- 9.1 ¿Por qué compilar aplicaciones escalables y confiables? 124
 - Escalado de las VM verticalmente 125 ▪ Escalado de aplicaciones web verticalmente 127 Escalado de recursos horizontalmente 128*
- 9.2 Conjuntos de escalado de máquinas virtuales 129
 - Creación de un conjunto de escalado de máquina virtual 131 ▪ Creación de reglas de escalado automático 133*
- 9.3 Escalado de una aplicación web 136
- 9.4 Laboratorio: instalación de aplicaciones en el conjunto de escalado o la aplicación web 139
 - Conjuntos de escalado de máquinas virtuales 139 ▪ Aplicaciones web 140*

10 Bases de datos globales con Cosmos DB 141

- 10.1 ¿Qué es Cosmos DB? 141
 - Bases de datos estructurados (SQL) 142 ▪ Bases de datos (NoSQL) no estructurados 142 ▪ Escalado de bases de datos 143 ▪ Cosmos DB se encarga de todo 144*
- 10.2 Creación de una cuenta y base de datos de Cosmos DB 145
 - Creación y relleno de una base de datos de Cosmos DB 145 Adición de redundancia global a una base de datos de Cosmos DB 149*
- 10.3 Acceso a los datos distribuidos globalmente 152
- 10.4 Laboratorio: implementación de una aplicación web que usa Cosmos DB 156

11 Administración del tráfico de redes y enrutamiento 158

- 11.1 ¿Qué es Azure DNS? 158
- 11.2 Delegación de un dominio real a Azure DNS 160
- 11.3 Enrutamiento global y resolución con el Administrador de tráfico 162

	<i>Creación de perfiles de Traffic Manager 164 ▪ Distribución global del tráfico a la instancia más cercana 167</i>
11.4	Laboratorio: implementación de aplicaciones web para ver Traffic Manager en acción 174
12	Monitoreo y solución de problemas 175
12.1	Diagnóstico de inicio de VM 175
12.2	Métricas y alertas de rendimiento 178 <i>Visualización de métricas de rendimiento con la extensión del diagnóstico de VM 178 ▪ Creación de alertas para condiciones de rendimiento 181</i>
12.3	Azure Network Watcher 182 <i>Verificación de flujos de IP 183 ▪ Visualización de reglas efectivas de NSG 184 Captura de paquetes de red 186</i>
12.4	Laboratorio: creación de alertas de rendimiento 188
PARTE 3 SEGURO POR DEFECTO	189
13	Copias de seguridad, recuperación y replicación 191
13.1	Azure Backup 191 <i>Directivas y retención 193 ▪ Programaciones de copia de seguridad 196 Restauración de VM 198</i>
13.2	Azure Site Recovery 201
13.3	Laboratorio: configuración de una VM para Site Recovery 204
14	Cifrado de datos 206
14.1	¿Qué es el cifrado de datos? 206
14.2	Cifrado en reposo 208
14.3	Storage Service Encryption 209
14.4	Cifrado de VM 211 <i>Almacenamiento de claves de cifrado en Azure Key Vault 211 ▪ Cifrado de una VM de Azure 213</i>
14.5	Laboratorio: cifrado de una VM 214
15	Protección de la información con Azure Key Vault 216
15.1	Protección de la información en la nube 216 <i>Almacenes de software y módulos de seguridad de hardware 217 Creación de un almacén de claves y secreto 219</i>

- 15.2 Identidades administradas para recursos de Azure 221
- 15.3 Obtención de un secreto dentro de una máquina virtual con identidad de servicio administrado 224
- 15.4 Creación e inyección de certificados 229
- 15.5 Laboratorio: configuración de un servidor web seguro 232

16 Azure Security Center y actualizaciones 234

- 16.1 Azure Security Center 234
- 16.2 Acceso Just-In-Time 237
- 16.3 Azure Update Management 241
 - Servicios combinados de administración de Azure 243 ▪ Revisión y aplicación de actualizaciones 245*
- 16.4 Laboratorio: habilitación de JIT y actualizaciones para una VM de Windows 249

PARTE 4 ASPECTOS INTERESANTES 251

17 Machine learning e inteligencia artificial 253

- 17.1 Descripción y relación de IA y ML 254
 - Inteligencia artificial 254 ▪ Machine learning 255*
 - La unión de IA y ML 256 ▪ Herramientas de Azure ML para científicos de datos 257*
- 17.2 Azure Cognitive Services 259
- 17.3 Creación de un bot inteligente para ayudar con pedidos de pizza 260
 - Creación de un bot de Azure Web App 260 ▪ Lenguaje e intención de comprensión con LUIS 261 ▪ Creación y ejecución de un bot de aplicación web con LUIS 264*
- 17.4 Laboratorio: adición de canales para la comunicación de un bot 267

18 Azure Automation 269

- 18.1 ¿Qué es Azure Automation? 269
 - Creación de una cuenta de Azure Automation 271 ▪ Activos y runbooks de Azure Automation 272*

- 18.2 Runbook de ejemplo de Azure Automation 274
Ejecución y visualización de la salida de un runbook de ejemplo 276
- 18.3 Desired State Configuration (DSC) de PowerShell 278
Definición y uso de DSC de PowerShell y un servidor de extracción de Azure Automation 280
- 18.4 Laboratorio: uso de DSC con Linux 282

19 Contenedores de Azure 284

- 19.1 ¿Qué son los contenedores? 284
- 19.2 El enfoque de los microservicios a las aplicaciones 287
- 19.3 Azure Container Instances 289
- 19.4 Azure Kubernetes Service 293
Creación de un clúster con Azure Kubernetes Services 294
Ejecución de un sitio web básico en Kubernetes 295
- 19.5 Laboratorio: escalado de sus implementaciones de Kubernetes 298

20 Azure y la Internet de las Cosas 300

- 20.1 ¿Qué es la Internet de las Cosas? 300
- 20.2 Administración centralizada de dispositivos con Azure IoT Hub 303
- 20.3 Creación de un dispositivo Raspberry Pi simulado 306
- 20.4 Transmisión de datos de Azure IoT Hub a las aplicaciones web de Azure 309
- 20.5 Revisión de componentes de Azure IoT 315
- 20.6 Laboratorio: exploración de casos prácticos para IoT 316

21 Informática sin servidor 317

- 21.1 ¿Qué es la informática sin servidor? 317
- 21.2 Plataformas de mensajes Azure 319
Azure Event Grid 320 Azure Event Hubs y Service Bus 321
Creación de un Service Bus y su integración con un IoT Hub 322
- 21.21.4 Creación de una Azure Function App para analizar datos del dispositivo de IoT 328
- 21.5 No deje de aprender 332
Materiales de aprendizaje adicionales 333 ▪
Recursos de GitHub 333 *Reflexión final* 333

prólogo

Esta segunda edición de *Aprenda Azure en un mes de almuerzos* me recuerda que las cosas cambian con rapidez y que siempre hay que seguir aprendiendo. Atrás quedaron los días en los que se podía hacer un curso de una semana de duración sobre Windows Server y ejecutarlo cómodamente durante años sin cambiar mucho. Esto no significa que el mundo de la TI sea un lugar más aterrador, pero es necesario abordar la informática en la nube con una mente abierta y estar dispuesto a adaptarse de manera constante.

Cuando comencé a trabajar con Azure, el número de servicios disponibles era casi abrumador. Sabía que debía prestar atención a la seguridad, el rendimiento, la redundancia y la escala, pero no sabía cómo adaptar más de una década de administración de servidores a gran escala en el mundo de la informática en la nube. Con el tiempo, comencé a aprender sobre los diversos servicios de Azure que proporcionan esos componentes clave. Estos servicios pocas veces funcionan de forma aislada, pero no sabía cuál era la mejor manera de integrarlos o cómo decidir qué servicio utilizar para cada tarea. Este libro es una manera de explicar a mi yo del pasado y a muchos otros que siguieron un camino similar, cómo entender rápidamente los servicios básicos de Azure y hacerlos funcionar juntos.

Este libro tiene más de 350 páginas, ¡pero apenas descubre la superficie de lo que se puede hacer en Azure! Para ayudar a darle una comprensión sólida de los conceptos necesarios para tener éxito a medida que crea soluciones en Azure, tuve que elegir sobre qué temas escribir. El libro no abarca todos los 100 o más servicios de Azure, y no entra en detalles exhaustivos sobre los servicios incluidos. En cambio, se enfoca en las áreas clave de algunos de los servicios principales y muestra ejemplos para conectar de forma segura todo y presenta posibilidades de lo que se puede crear en Azure.

La informática en la nube está cambiando continuamente. No hay ciclos de versiones cada tres o cuatro años, o grandes implementaciones de actualizaciones. Creo que hoy es un excelente momento para crear soluciones y escribir código; siempre hay una oportunidad de aprender algo nuevo y mejorar. Espero que aprenda a ejecutar aplicaciones geniales en Azure y disfrute explorando todos los servicios disponibles.

agradecimientos

Muchas personas de Manning Publications tras bambalinas ayudaron a publicar este libro. Agradezco especialmente a Mike Stephens por tener la visión inicial para comenzar este proyecto. Agradezco al editor, Marjan Bace, y a todos los integrantes de los equipos editoriales y de producción. Agradezco a los revisores técnicos de pares dirigidos por Aleksandar Dragosavljevic: Ariel Gamino, Charles Lam, Ernesto Cardenas Cangahuala, George Onofrei, Glen Thompson, Jose Apablaza, Juraj Borza, Michael Langdon, Michael Wall, Peter Kreyenhop, Rick Oller, Rob Ruetsch, Robert Walsh y Vishal Singh. Y por último, en el aspecto técnico, gracias a Karsten Strøbaek, quien actuó como editor y corrector técnico del libro.

Para esta segunda edición, agradezco enormemente a Phil Evans y Davanand Bahall por su apoyo y la libertad de actualizar este libro. Se trataba de un proyecto a deshoras, fuera de mi trabajo diario en Microsoft, pero a mucha gente le entusiasmó y le afectó. Gracias a David Tolkov y Tim Teebken, que me dieron la oportunidad de convertirme en alguien capaz de escribir este libro. Y mira, Jean-Paul Connock, ¡hemos ganado una Copa Stanley desde la última vez! ¡Vivan los Blues!

Gracias a Rick Claus por apoyar la necesidad de una documentación técnica sólida sobre Azure, y a Marsh Macy y Neil Peterson por su apoyo personal y orientación en la redacción de la versión original de este libro. Todavía tenemos que empezar con ese autobús escolar.

acerca de este libro

Este libro está diseñado para darle una base concreta para tener éxito como desarrollador o ingeniero de TI en Azure. Aprenderá acerca de las soluciones de Infraestructura como servicio (IaaS) y Plataforma como servicio (PaaS), además de cuándo utilizar cada enfoque. A medida que avance por los capítulos, aprenderá a planificar adecuadamente para la disponibilidad y escala, a tener en cuenta la seguridad y considerar el costo y el rendimiento. Para el final del libro, debería poder integrar las próximas tecnologías, como contenedores y Kubernetes, inteligencia artificial y machine learning (IA+ ML), e Internet de las Cosas (IoT).

Con respecto a cómo se crean y ejecutan sus aplicaciones y servicios, Azure le permite elegir el sistema operativo, las herramientas de la aplicación y la plataforma con la que se sienta más cómodo. Este libro trata especialmente las tecnologías que no son de Microsoft, como Linux, Python y Node.js. Los ejemplos de comandos usan la CLI de Azure, no Azure PowerShell. Estas fueron decisiones conscientes para mostrarle que usar Azure no significa que usar Windows Server, IIS o ASP.NET.

Al trabajar en la nube, a menudo se trabaja en distintas plataformas y se deben aprender nuevos temas, que es otra razón para mostrar las tecnologías y plataformas que no son de Microsoft. Quería presentarle algunas de estas nuevas áreas antes de que se las encuentre en el mundo real. A lo largo del libro, le enseñaré los conceptos y pasos necesarios para integrar los servicios de Azure, con el fin de que pueda cambiar de plataforma o lenguaje según desee y tenga los mismos conocimientos para aplicar.

Hoja de ruta

El libro está organizado en 4 partes y 21 capítulos:

- La parte 1 abarca algunos de los servicios principales de infraestructura y plataforma Azure: máquinas virtuales, aplicaciones web, almacenamiento y redes.

- La parte 2 detalla cómo proporcionar alta disponibilidad y redundancia: plantillas, zonas y conjuntos de disponibilidad, equilibradores de carga, escalado automático, bases de datos distribuidas y enrutamiento de tráfico. Para el final del capítulo 12, debería tener conocimientos sólidos sobre cómo crear aplicaciones de alto rendimiento y distribuidas en Azure.
- La parte 3 abarca aspectos de seguridad, como las copias de seguridad y la recuperación, el cifrado, la administración de claves digitales y las actualizaciones. Cuando haya completado el capítulo 16, estará en camino para crear aplicaciones estables en Azure.
- Para terminar el libro, la parte 4 presenta un poco de diversión, gracias a la exploración de nuevas áreas de informática, como la informática sin servidor y las aplicaciones basadas en contenedores. Estos capítulos presentan áreas de Azure que le dan una visión de cómo podría ser el futuro de las aplicaciones de producción.

Excepto en la parte 4, que se llama acertadamente "Aspectos interesantes", debe intentar leer los capítulos del libro en orden. No trabajará en el mismo proyecto en capítulos sucesivos, pero cada capítulo se basa en una teoría anterior y en ejemplos del laboratorio práctico.

El Capítulo 1 lo guía para crear una cuenta de prueba gratuita en Azure, que es suficiente para completar los ejercicios del laboratorio práctico en cada capítulo. También se proporciona un poco más de antecedentes de Azure y cómo encontrar ayuda adicional a lo largo del camino. Mencione esta página web algunas veces en el libro (tal vez estoy un poco sesgado), pero <http://docs.microsoft.com/azure> es el mejor lugar para encontrar documentación adicional y ayuda en cualquier área de Azure que le interese.

Acerca de los ejemplos y el código fuente

Este libro contiene muchos ejemplos de código fuente, tanto en listados numerados como en línea con el texto normal. En ambos casos, el código fuente está en un formato de fuente de ancho fijo como esta para separarla del texto normal.

En muchos casos, hemos cambiado el formato del código fuente original con saltos de línea y sangrías modificadas para ajustar el espacio disponible en la página del libro. En algunos casos, ni siquiera esto fue suficiente, y los listados incluyen marcadores de continuación de línea (➡). Además, eliminamos los comentarios en el código fuente de los listados cuando el código se describe en el texto. Hay anotaciones en el código en muchos de los listados, que destacan conceptos importantes.

El código fuente de este libro, junto con los scripts, las plantillas y los recursos de ayuda que lo acompañan, se pueden encontrar en <https://www.manning.com/books/learn-azure-in-a-month-of-lunches-second-edition> y en el repositorio de GitHub del libro (<https://github.com/fouldsy/azure-mol-samples-2nd-ed>).

Todos los ejercicios prácticos pueden completarse en Azure Portal y con Azure Cloud Shell, una shell interactiva basada en el navegador para la CLI de Azure y Azure PowerShell. No es necesario instalar herramientas en su máquina, y puede utilizar cualquier equipo y sistema operativo que desee, siempre y cuando sea compatible con un navegador web moderno.

Azure Portal suele implementar cambios menores. Parte del desafío de utilizar cualquier servicio en la nube es que las cosas pueden ser un poco diferentes de lo que eran el día anterior.

En esta segunda edición del libro se intenta minimizar el número de capturas de pantalla del portal, pero no se preocupe si lo que ve es ligeramente diferente de lo que se muestra en el libro. Los parámetros necesarios suelen ser los mismos, pero el diseño puede ser diferente. Si hay nuevas opciones en el portal que no utilizo específicamente en un ejercicio o laboratorio, lo habitual es que sea seguro aceptar los valores predeterminados que se proporcionan.

Si trabaja fuera de Azure Cloud Shell, tenga cuidado con los ejemplos de comandos. Las shells basadas en Windows, como PowerShell y CMD, tratan los saltos de línea y las continuaciones de forma diferente de las shells basadas en *nix, como Azure Cloud Shell. Muchos de los ejemplos de comandos se ejecutan en varias líneas. Los comandos se muestran con un carácter de barra invertida (\) para indicar que el comando continúa en la línea siguiente, como en este ejemplo:

```
az resource group create \
--name azuremol \
--location eastus
```

No tiene que escribir esos caracteres de barra invertida, pero si lo hace, los comandos largos pueden ser más legibles en la pantalla. Si decide trabajar localmente en su equipo con una shell de Windows, puede utilizar una comilla invertida (^) en lugar de una barra invertida. Por ejemplo, en una shell de PowerShell o CMD con Python para Windows instalado, cambie el comando anterior de la siguiente manera:

```
az resource group create ^
--name azuremol ^
--location eastus
```

Al principio, esta convención puede parecer confusa, pero la sigo en el libro porque la documentación oficial en <https://docs.microsoft.com/azure> utiliza este formato. Los comandos de la CLI de Azure, que son los que usamos principalmente en este libro, suponen una shell basada en *nix y, por lo tanto, usan un carácter de barra invertida. Los comandos de Azure PowerShell suponen una shell basada en Windows y, por lo tanto, utilizan una comilla invertida. Esta diferencia de comportamiento cobrará sentido rápidamente, y descubrirá que es fácil pasar de una shell a otra. Si es nuevo en el trabajo en plataformas, esta diferencia puede ser un truco divertido de aprender.

Le recomiendo que eche un vistazo al Subsistema de Windows para Linux (WSL) si ejecuta Windows 10 y quiere sumergirse en la CLI de Azure y en los sistemas basados en *nix en general; puede obtener información en <https://docs.microsoft.com/windows/wsl>. WSL, y las últimas mejoras en WSL2 le brindan una experiencia nativa del kernel de Linux mientras ejecuta Windows. No intente hacerse a la idea demasiado. Solo tiene que saber que puede ejecutar comandos y aplicaciones nativas de Linux sin preocuparse por los diferentes saltos de línea o definiciones de variables. Para que se quede con la boca abierta, PowerShell está disponible para .NET Core, que también se ejecuta en Linux. Puede ejecutar PowerShell en Linux en Windows.

Foro de discusión de liveBook

La compra de *Aprenda Azure en un mes de almuerzos* incluye acceso gratuito a un foro web privado dirigido por Manning Publications, donde puede dejar comentarios sobre el libro, hacer preguntas técnicas, y recibir ayuda del autor y de otros usuarios. Para

obtener acceso al foro, diríjase a <https://livebook.manning.com/book/learn-azure-in-a-month-of-lunches-second-edition/discussion>. También puede obtener más información sobre los foros de Manning y las normas de conducta en <https://livebook.manning.com/discussion>.

El compromiso de Manning con nuestros lectores es proporcionar un lugar donde pueda tener lugar un diálogo significativo entre los lectores individuales y entre los lectores y el autor. No es un compromiso con ninguna cantidad específica de participación por parte del autor, cuya contribución al foro sigue siendo voluntaria (y no remunerada). Le recomendamos que intente realizarle preguntas interesantes para mantener su interés. El foro y los archivos de discusiones anteriores serán accesibles desde el sitio web del editor siempre y cuando el libro esté impreso.

acerca del autor

IAIN FOULDS es un desarrollador de contenidos sénior en Microsoft y actualmente escribe documentación técnica para Azure Active Directory. Antes, fue ingeniero principal de campo con Microsoft para tecnologías de virtualización, como Azure, Hyper-V y System Center Virtual Machine Manager. Con más de 15 años de experiencia en TI, la mayor parte de ellos en operaciones y servicios, Iain adoptó la virtualización muy temprano con VMWare y ha ayudado a crear y enseñar a otros sobre la informática en la nube durante años.

Originario de Inglaterra, ha vivido en Estados Unidos por más de una década y hoy en día reside en las afueras de Seattle con su esposa y sus dos hijos pequeños, a quienes dedicó este libro. Es fanático del fútbol (por desgracia, llamado "soccer" donde vive) y también disfruta del hockey sobre hielo, además de casi cualquier forma de carreras a motor. Fuera de la informática, sus intereses incluyen los autos clásicos y de carrera, la fotografía de aviación y tocar la guitarra. También es un gran admirador del modelismo ferroviario, por lo que asiste periódicamente a espectáculos y eventos en todo el noroeste del Pacífico donde también es voluntario.

Parte 1

Servicios principales de Azure

P

ara compilar la próxima gran aplicación, necesita una sólida comprensión de los recursos básicos en Azure. Temas como el almacenamiento y la red pueden no ser lo más interesante, pero son fundamentales para mucho de lo que se ejecuta en Azure. Antes de que pueda empezar a meterse en máquinas virtuales redundantes, de varias instancias o en las aplicaciones web de Azure, resulta útil ver las opciones disponibles y las tareas de administración para una sola instancia. Este enfoque le permite aprender acerca de las diferencias y similitudes entre el enfoque IaaS de las VM y el enfoque PaaS de las aplicaciones web. En los capítulos 1 a 5, exploramos las VM, las aplicaciones web, el almacenamiento básico y las funcionalidades de redes virtuales.

Antes de comenzar



Azure es uno de los mayores proveedores de nube informática de uso público para servicios como máquinas virtuales (VM), contenedores, informática sin servidor y machine learning. No vamos a ver en detalle los 100 o más servicios de Azure en este libro, pero aprenderá acerca de los servicios y funciones básicos que abarcan la mayor parte de lo que necesita para empezar a compilar y ejecutar soluciones en Azure. Veremos un ejemplo común de cómo compilar y ejecutar una aplicación web, y podrá ver cómo utilizar algunos de los servicios de infraestructura y plataforma principales que pueden facilitar su trabajo.

Con Azure, no necesita una varita mágica para predecir cuántos servidores o cuánto almacenamiento necesitará en los próximos tres años. No espere más mientras obtiene la aprobación del presupuesto, reciba el envío del nuevo hardware y luego apile, instale y configure todo. No necesita preocuparse acerca de qué versiones de software o bibliotecas están instaladas mientras escribe su código.

Simplemente seleccione un botón y cree cualquier recurso que se necesite. Solo se paga por cada minuto que se ejecutan esos recursos, o por la cantidad de espacio de almacenamiento o ancho de banda de red utilizados. Cuando ya no necesite los recursos, puede apagarlos o eliminarlos. Y si de repente necesita aumentar la cantidad de potencia de proceso por un factor de 10, seleccione un botón, espere un par de minutos, y ahí estará. Otra persona administra todo esto, lo que le permite centrarse de lleno en sus aplicaciones y clientes.

1.1 *¿Es este el libro que busca?*

La industria de TI está en un período algo transitorio cuando se trata de cargos de trabajo. Usted puede referirse a sí mismo como un profesional de TI, un desarrollador de software, un administrador de sistemas o un ingeniero

de operaciones y desarrollo. Independientemente de cómo se llame, si quiere aprender las habilidades básicas necesarias para compilar y ejecutar aplicaciones seguras y altamente disponibles en la nube, está en el lugar correcto. En términos genéricos, probablemente se pueda clasificar dentro de operaciones o desarrollo de TI, pero la verdad es que estas actividades se entrelazan bastante, especialmente en la informática en la nube. Ya sea en desarrollo o en operaciones, es importante comprender la infraestructura básica y los servicios de plataforma para compilar y ejecutar las aplicaciones que mejor sirvan a sus clientes.

En esta segunda edición del libro, se presentan algunos de estos conceptos básicos en Azure y se enseñan las habilidades que necesita para tomar decisiones fundamentadas. Al entrar en este libro, debe tener algo de experiencia previa con las VM y conocer los fundamentos de la red y de almacenamiento. También debiera ser capaz de crear un sitio web básico, además de entender lo que son un certificado SSL y una base de datos. Después de cubrir los procesos básicos, echaremos un vistazo rápido a las nuevas y futuras tecnologías. Quiere estar a la vanguardia de dónde puede llevarlo su trabajo, así que aprenderá sobre contenedores, la Internet de las Cosas, el machine learning, la inteligencia artificial y la informática sin servidor. Tanto los que se autodenominan "desarrolladores" como los "profesionales de TI" encontrarán interesantísimas áreas nuevas para aprender.

1.2 **Cómo usar este libro**

Me gustan los sándwiches, así que el almuerzo es un buen momento para jugar con fascinante tecnología nueva. Usted puede ser un amante de la noche que tiene un poco de tiempo extra por las tardes, o le puede gustar madrugar, y así aprovechar de trabajar un capítulo durante el desayuno. No hay tiempo correcto o incorrecto para aprender, pero si pudiera reservar unos 45 minutos, debiera ser capaz de leer un capítulo y de completar los ejercicios. Cada capítulo cubre algo nuevo, así que dese tiempo para incorporar la lección de cada día.

1.2.1 **Los capítulos principales**

El libro se divide en cuatro partes, lo que resulta conveniente si considera que hay cuatro semanas en un mes:

- La parte 1 (capítulos 1 a 5) cubre algunos de los recursos básicos de Azure. Intente seguir estos capítulos en orden para lograr un entendimiento sólido. Luego, puede centrarse en los otros capítulos que más le llamaron la atención.
- La parte 2 (capítulos 6 a 12) cubre la disponibilidad y la escalabilidad. Aprenderá a escalar automáticamente los recursos para aumentarlos o disminuirlos, a equilibrar la carga del tráfico y a controlar los eventos de mantenimiento sin tiempos de inactividad. Si desea obtener información sobre cómo ejecutar aplicaciones de alta disponibilidad a escala global, esta parte es para usted.
- La parte 3 (capítulos 13 a 16) es para los geeks de la seguridad. Cubre temas como cifrar VM, almacenar certificados SSL en un almacén seguro, hacer copias de seguridad y recuperar datos.
- La parte 4 (capítulos 17 a 21) cubre una mezcla de áreas fascinantes para darle una idea de lo que Azure puede hacer por usted y sus clientes. Estudiaremos automatización, contenedores, Internet de las Cosas e informática sin servidor. Escoja algo de su interés y ¡diviértase!

1.2.2 **Pruébelo ahora**

¿Solo quiere leer o quiere subirse las mangas y jugar con Azure? A lo largo del libro, encontrará pequeñas tareas que le permitirán probar rápidamente algo nuevo. Si tiene tiempo, pruébelas. La mayor parte del tiempo de práctica viene en un ejercicio de laboratorio al final del capítulo, pero leer detenidamente e ir probando nuevos conceptos a lo largo del camino puede resultar muy valioso. Algunos de estos ejercicios lo guían paso a paso; otros lo harán pensar un poco más y aprenderá cómo desarrollar soluciones por su cuenta como lo haría en el mundo real.

1.2.3 **Laboratorios prácticos**

Cada capítulo finaliza con un ejercicio práctico de laboratorio. Algunos capítulos, como este, tienen un ejercicio de laboratorio al medio del capítulo. Estos ejercicios de laboratorio son donde aprenderá cómo se unen todas las piezas de Azure y así empezar a compilar cierta memoria muscular mental. Agarre el teclado y el mouse, y empiece a compilar algo impresionante.

1.2.4 **Código fuente y materiales suplementarios**

El código fuente de este libro, junto con los scripts, las plantillas y los recursos de ayuda que lo acompañan, se pueden encontrar en <https://www.manning.com/books/learn-azure-in-a-month-of-lunches-second-edition> y en el repositorio de GitHub en <https://github.com/fouldsy/azure-mol-samples-2nd-ed>. Además, puede participar en el foro del libro en <https://livebook.manning.com/book/learn-azure-in-a-month-of-lunches-second-edition/discussion>.

1.3 **Creación de su entorno de laboratorio**

Este libro no es denso en cuanto a conceptos y arquitectura, pero sí en cuanto a tiempo práctico con la plataforma Azure. Para obtener esta práctica, necesita una cuenta de Azure.

1.3.1 **Cómo crear una cuenta gratuita de Azure**

Azure ofrece una cuenta de prueba gratuita que dura 30 días y proporciona hasta USD 200 de crédito gratuito. Este crédito debería ser suficiente para que pueda completar todos los capítulos y ejercicios, otorgándole espacio para explorar un poco y divertirse en el camino. Muchos servicios y funciones de Azure nunca tienen costo, incluso después de que finalice el período de prueba.

Pruébelo ahora

Siga los pasos de esta sección para crear su cuenta de Azure gratuita:

- 1 Abra su navegador web, vaya a <https://azure.microsoft.com/free> y seleccione la opción para comenzar con una cuenta gratuita de Azure.
- 2 Cuando se le indique, inicie sesión en su cuenta de Microsoft. Si necesita una cuenta de Microsoft o desea crear una nueva cuenta, elija el vínculo Crear una nueva cuenta de Microsoft.

- 3 Cuando haya iniciado sesión en la cuenta de Microsoft, complete las notificaciones para crear una cuenta de Azure gratuita:
 - Ingrese sus datos personales como se le pide.
 - Para ayudar a minimizar el abuso y el fraude, proporcione un número de teléfono para verificar su identidad mediante un mensaje de texto o una llamada telefónica.
 - También se requiere una tarjeta de crédito para verificar la identidad, pero aquí no se realizará cargo alguno. Su cuenta no comienza a facturarse hasta después de 30 días o cuando haya utilizado su crédito gratuito de USD 200. No pasará automáticamente a una suscripción del tipo "pago por uso" al final del período de prueba. Es posible que vea una pequeña retención de verificación de USD 1 (o su equivalente en moneda local) que se reembolsa en unos días.
- 4 Revise y acepte el acuerdo de suscripción y la política de privacidad de Azure y, a continuación, seleccione Registrarse. Su suscripción en Azure puede tomar unos minutos en estar lista.
- 5 Cuando finalice el proceso de inscripción y se cargue Azure Portal, tome la visita rápida para aprender a navegar.

Su panel (la página principal del portal) se ve vacío en este momento. Pero en el capítulo 2 se sumergirá en la creación de su primera VM y comenzará a parecerse a la figura 1.1.

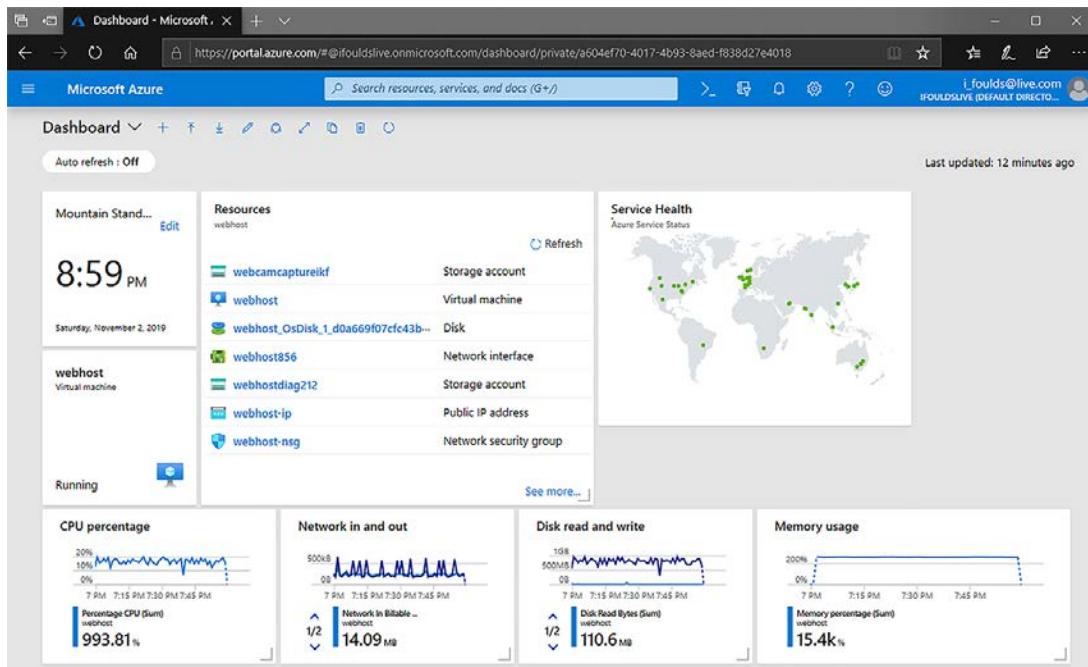


Figura 1.1 Azure Portal está listo para que cree sus propias aplicaciones y soluciones

¿Es verdaderamente gratis?

Azure tiene un Marketplace que contiene cientos de imágenes precompiladas (la base de las VM) y soluciones que puede implementar. Usaremos algunas de estas ofertas del Marketplace a lo largo del libro, son maneras muy convenientes de implementar rápidamente un conjunto de aplicaciones.

No todo este contenido de Azure Marketplace es gratuito: algunos editores externos combinan los costos de licencias o de soporte en la solución que usted implemente. Por ejemplo, una VM que implemente desde Red Hat puede incurrir en una tarifa adicional que cubra el acuerdo de soporte y la licencia de Red Hat. Estos cargos no están cubiertos por su crédito gratuito de prueba; solo se cubre el uso de la VM base.

Los ejercicios en este libro solo utilizan los recursos aptos para prueba gratuita. Pero si comienza a explorar otras ofertas interesantes en el Marketplace de Azure, preste atención a lo que compila. Cualquier solución que incluya tarifas adicionales debe explicarlas claramente antes de su implementación.

1.3.2 Ejercicio de laboratorio adicional: creación de una cuenta gratuita de GitHub

GitHub es un servicio web gratuito que muchas organizaciones y personas utilizan para administrar proyectos como código, plantillas y documentación. Azure tiene cientos de plantillas gratuitas y ejemplos de scripts a los que puede contribuir y también utilizar. Esta es una de las ventajas de la comunidad open source: compartir y devolver la mano a los demás.

Algunos de los ejercicios de este libro usan recursos de GitHub. No necesita una cuenta de GitHub para hacer nada de esto, pero si no tiene una cuenta, no podrá guardar ninguna modificación ni comenzar a crear su propia colección de plantillas y scripts. Crear una cuenta de GitHub es una parte opcional (pero sumamente recomendable) para compilar su entorno de laboratorio:

- 1 Abra su navegador web y vaya a <https://github.com> .
- 2 Para crear una cuenta de GitHub gratuita, escriba un nombre de usuario, una dirección de correo electrónico y una contraseña. Recibirá un mensaje de validación de GitHub.
- 3 Seleccione el vínculo en el correo electrónico de validación para activar su cuenta.
- 4 Consulte algunos de los repositorios Azure que proporcionan recursos de ejemplo:
 - Plantillas de inicio rápido de Azure: <https://github.com/Azure/azure-quickstart-templates>
 - CLI de Azure: <https://github.com/Azure/azure-cli>
 - Utilidades de Azure DevOps: <https://github.com/Azure/azure-devops-utils>
 - Recursos del libro *Aprenda Azure en un mes de almuerzos*: <https://github.com/fouldsy/azure-mol-samples-2nd-ed>

1.4 Un poco de ayuda

Este libro no puede cubrir todo lo que Azure ofrece. Aunque lo intentara, para cuando haya terminado de leer este capítulo, seguro que habrá algo nuevo en Azure. La informática en la nube avanza rápidamente y siempre se están lanzando nuevos

servicios y funcionalidades. Puede que sea un poco sesgado, pero a medida que comience a explorar Azure y quiera aprender acerca de los servicios adicionales, el excelente sitio <https://docs.microsoft.com/azure> es el mejor lugar para empezar. Cada servicio Azure está documentado con ejemplos de inicio rápido, tutoriales, ejemplos de código, referencias de desarrolladores y guías de arquitectura. También puede acceder a las opciones de soporte gratuito y pagado si necesita ayuda en el camino.

1.5

Comprendión de la plataforma Azure

Antes de adentrarse en el resto de este libro, vamos a dar un paso atrás y cubrir lo que es Azure y los servicios que están disponibles. Como mencionamos anteriormente, Azure es un proveedor de informática en la nube de escala mundial. En el momento de la redacción de este libro, existen 54 regiones de Azure. Cada región contiene uno o más centros de datos. En comparación, los otros dos principales proveedores de nube operan en 23 regiones (Amazon Web Services [AWS]) y 20 regiones (Google Cloud).

La informática en la nube proporciona mucho más que procesos. Azure tiene más de 100 servicios, agrupados en familias de servicios relacionados, tales como informática, web + móvil, contenedores, e identidad. Con todos estos servicios, Azure cubre muchos modelos de servicio. Tomemos un trozo de pizza para el almuerzo y aprovechemos de entender lo que esto significa (figura 1.2).



Figura 1.2 La pizza como modelo de servicio. Cuando pasa de la pizza casera, donde usted se pone con todo, al modelo de restaurante, donde solo debe llegar al lugar, las responsabilidades y las demandas de gestión también cambian.

En la pizza como modelo de servicio, dispone de cuatro opciones para elegir. A medida que avanza a través de los modelos, se preocupa cada vez menos por el proceso de comer un trozo de pizza:

- *Hecha en casa:* hace la masa; añade la salsa, los ingredientes y el queso; hornea la pizza en el horno; compra bebidas y se sienta a comer en su mesa.

- *Llevary hornear:* compra una pizza lista. Solo tiene que hornearla, comprar bebidas y sentarse a comer en su mesa.
- *Entrega a domicilio:* pide una pizza que se le entrega en su casa. Solo tiene que comprar bebidas y sentarse a comer en su mesa.
- *Restaurante:* quiere salir y comer su pizza haciendo un mínimo esfuerzo.

Ahora que tiene hambre, echemos un vistazo al modelo más tradicional que involucra algunos procesos (figura 1.3). Este modelo se ve más parecido a lo que vería en Azure.

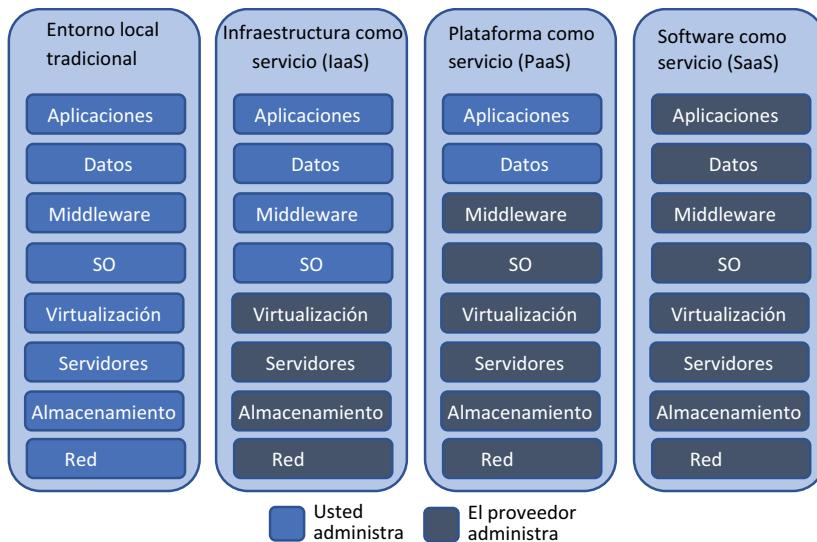


Figura 1.3 Modelo de servicio de informática en la nube

A medida que avanza a través de los modelos, gestiona menos recursos subyacentes y puede destinar más tiempo y energía a sus clientes:

- *Local:* configura y administra todo el centro de datos, como los cables de red, el almacenamiento y los servidores. Usted es responsable de todas las partes del entorno de la aplicación, el soporte y la redundancia. Este enfoque proporciona el máximo control, pero involucra una gran cantidad de tareas administrativas.
- *Infraestructura como servicio (IaaS):* usted adquiere los recursos de procesos base de un proveedor que administra la infraestructura básica. Crea y administra las VM, los datos y las aplicaciones. El proveedor de la nube es responsable de la infraestructura física, la administración de hosts y la resiliencia. Es posible que todavía tenga un equipo de infraestructura para apoyar e implementar las VM, pero el equipo no tiene que lidiar con la dedicación ni los costos de administrar el equipo físico.

Este enfoque es bueno cuando comienza migrar las aplicaciones fuera de su propio entorno local. La administración y las operaciones suelen ser similares a un entorno dentro de las instalaciones, así que la IaaS proporciona una progresión natural para que el negocio, TI, y los dueños de la aplicación se sientan cómodos con la nube.

- *Plataforma como servicio (PaaS)*: usted adquiere la plataforma subyacente de un proveedor que administra el sistema operativo y los parches, y trae sus aplicaciones y datos. No se preocupe por las VM o la red virtual; además su equipo de operaciones puede dedicar más tiempo a la confiabilidad y el rendimiento de las aplicaciones.

Este enfoque suele comenzar a hacer que la organización de TI y el negocio se adapten y ejecuten las aplicaciones en la nube. Su enfoque está en las aplicaciones y sus clientes, con menos preocupaciones sobre la infraestructura necesaria para ejecutar esas aplicaciones.

- *Software como servicio (SaaS)*: usted solo necesita tener acceso al software y el proveedor proporciona todo lo demás. Los desarrolladores pueden compilar sobre una plataforma existente para proporcionar personalizaciones o funciones únicas, sin tener que mantener una gran base de código.

Este enfoque suele ser desalentador al principio, pero es probable que ya conozca y utilice útiles una oferta de SaaS exitosa, como Salesforce, Office 365 o el paquete de correo o documentos de Google. Usted utiliza correo electrónico, crea documentos o presentaciones, o administra la información de contacto de los clientes y la información de ventas. Su enfoque está en el contenido que crea y administra, no en cómo hacer que la aplicación se ejecute.

La mayor parte de lo que crea en Azure se clasifica en las áreas de IaaS y PaaS. Los principales casos de uso incluyen VM y redes virtuales (IaaS) o los servicios Azure Web Apps, Functions y Cosmos DB (PaaS) de Azure. Si es desarrollador, las soluciones PaaS son probablemente las áreas que más le interesen, ya que Microsoft cubre las partes de la infraestructura para que se pueda enfocar en su código. Los profesionales de TI pueden inclinarse más hacia las soluciones de IaaS para compilar y controlar la infraestructura Azure.

Nunca deje de aprender

No olvide que incluso cuando un negocio se cambia del modelo IaaS a PaaS, el profesional de TI sigue siendo esencial. Es importante entender lo que sucede debajo del nivel de PaaS cuando se diseña o se crea una solución. Si usted es un profesional de TI, no se salte los capítulos de las soluciones PaaS en Azure, puede sumar mucho valor para su negocio y sus clientes si entiende la transición a ese modelo de implementación.

1.5.1 Virtualización en Azure

La virtualización es la verdadera magia detrás de Azure. Los modelos IaaS, PaaS y SaaS utilizan la virtualización para hacer funcionar sus servicios. El concepto de virtualización no es nada nuevo, si nos remontamos a los días del sistema central de los años 60. A mediados de la década del 2000, la virtualización de servidores en los centros de datos comenzó a ganar impulso y ahora solo unas pocas cargas de trabajo se implementan en servidores sin sistema operativo en lugar de ser virtualizadas.

Hay libros que se dedican por entero a la virtualización, pero esta es una descripción rápida, la virtualización divide lógicamente los recursos físicos de un servidor en recursos virtuales a los que se puede acceder de forma segura mediante cargas de trabajo individuales. Una VM es uno de los recursos más comunes de la informática en la nube. Una VM contiene una CPU (vCPU), memoria (vRAM), almacenamiento (vDisk) y conectividad de red (vNIC) virtuales, como se muestra en la figura 1.4.

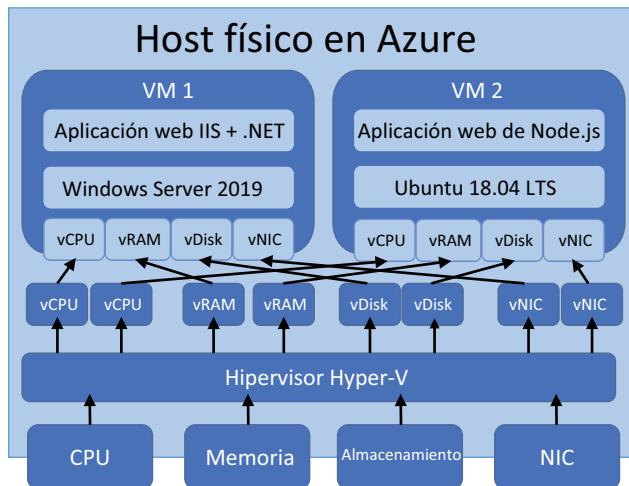


Figura 1.4 Virtualización en acción en un host físico en Azure

Además de los servidores físicos, es común virtualizar el almacenamiento y la creación de redes, lo que permite a la plataforma Azure definir rápidamente todo lo que necesita en software. No se requiere ninguna interacción física o configuración manual de los dispositivos. No tiene que esperar a que otro equipo proporcione una dirección IP, abra un puerto de red o agregue almacenamiento.

En esencia, es algo así como que Azure funciona en Windows. Una versión modificada del hipervisor de Hyper-V acciona los servidores de proceso. Hyper-V es un hipervisor tipo 1 (sin sistema operativo) que ha estado disponible en Windows Server durante una década. Y no se preocupe, ¡todavía puede ejecutar Linux como carga de trabajo de primera clase con soporte completo! Microsoft es un gran colaborador de la comunidad Linux y de kernel. Algunos de los núcleos de red definidos por software en Azure son impulsados por una solución personalizada basada en Debian Linux (software para redes abiertas en la nube [SONiC]) que Microsoft ha hecho open source. Puede realizar una visita virtual a los centros de datos de Microsoft en <https://azure.microsoft.com/global-infrastructure>.

1.5.2 Herramientas de administración

Azure ofrece demasiados servicios, ¿cómo utilizarlos? ¡Como quiera! Si desea seleccionar todo en un navegador web, existe un portal web impresionante. ¿Se siente cómodo utilizando PowerShell? Como es de esperar, hay un módulo Azure PowerShell.

También hay una herramienta de interfaz de línea de comandos (CLI) entre plataformas que es genial si usa MacOS o Linux. Y los desarrolladores pueden interactuar con Azure a través de las API REST utilizando una variedad de lenguajes comunes como .NET, Python y Node.js.

AZURE PORTAL

Azure Portal debe funcionar en cualquier navegador web moderno, y es una manera conveniente de usar Azure sin instalar nada en su equipo; además es ideal para aprender a crear y administrar recursos, ya que ve rápidamente una representación visual de todo.

Continuamente, se están agregando nuevas funciones y servicios a Azure, así que el portal puede cambiar levemente de lo que ve en las capturas de pantalla en este libro o en la documentación en línea y los blogs. El nombre de un botón puede cambiar un poco, o se puede agregar una nueva opción, pero las operaciones centrales siguen siendo las mismas. ¡Bienvenido al nuevo y valiente mundo de la informática en la nube!

AZURE CLOUD SHELL

Si quiere poner las manos en el teclado y escribir los comandos, el portal también incluye Azure Cloud Shell, que se muestra en la figura 1.5. Este shell es una consola interactiva en la web que proporciona un shell Bash, la CLI de Azure, y algunas herramientas de desarrollo de aplicaciones preinstaladas como Git y Maven. También hay una versión de PowerShell del Cloud Shell que, como su nombre lo indica, proporciona acceso a los últimos cmdlets de Azure PowerShell.

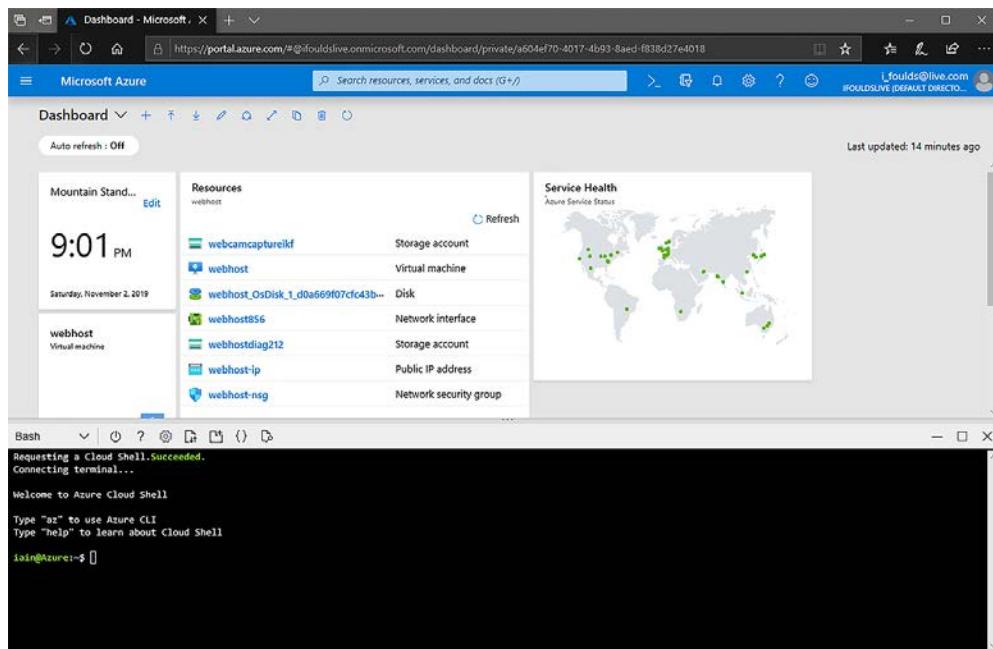


Figura 1.5 Azure Cloud Shell en el portal web

Puede acceder a Azure Cloud Shell desde un navegador web en cualquier equipo sin necesidad de instalar ninguna herramienta; para ello vaya a <https://shell.azure.com>. Los editores como Visual Studio Code (<https://code.visualstudio.com>) proporcionan acceso a Cloud Shell dentro de la aplicación. Hay incluso una aplicación Azure disponible para iOS y Android que le permite utilizar Azure Cloud Shell directamente desde su smartphone.

Con Azure Cloud Shell, siempre tendrá acceso a la última versión de las herramientas CLI o PowerShell. El almacenamiento permanente le permite crear y guardar scripts, plantillas y archivos de configuración.

HERRAMIENTAS LOCALES DE CLI DE AZURE Y POWERSHELL

Aunque Azure Cloud Shell tiene ventajas, a menudo es necesario acceder a sus sistemas de archivos y herramientas locales. Puede instalar la CLI de Azure o Azure PowerShell localmente para que pueda trabajar con recursos locales y recursos de Azure.

En este libro, utilizamos principalmente CLI de Azure (técticamente, CLI 2.0 de Azure). Puede parecer extraño elegirlo en lugar del PowerShell nativo de Microsoft. La ventaja es que los ejemplos y los ejercicios pueden funcionar tanto en Azure Cloud Shell como localmente en su equipo, independiente del sistema operativo que utilice. Aunque esta información no es parte de la configuración de su entorno de laboratorio, las siguientes guías detallan cómo instalar las herramientas de administración de Azure en su equipo:

- *Introducción a Azure PowerShell:* <https://docs.microsoft.com/powershell/azure/get-started-azureps>
- *Instalación de CLI de Azure:* <https://docs.microsoft.com/cli/azure/install-azure-cli>

Creación de una máquina virtual

¿Listo para ver lo rápido que se puede configurar un servidor web en Azure? En este capítulo, nos sumergiremos directamente en una de las solicitudes más comunes cuando se trata de VM: la creación de un servidor web básico. Esta carga de trabajo es un gran ejemplo de los componentes básicos de la infraestructura como servicio (IaaS) en Azure.

Suponga que usted trabaja en una pizzería que quiere expandir sus operaciones y aceptar pedidos en línea para entregar a domicilio o para llevar. Para generar presencia en línea, necesita un sitio web. En las primeras partes de este libro, exploraremos las diferentes funciones y servicios en Azure que le permiten compilar y ejecutar tanto las aplicaciones web IaaS como Plataforma como servicio (PaaS). Puede empezar a tomar decisiones informadas sobre cuándo compilar y ejecutar una VM para accionar un sitio web y cuándo puede usar PaaS para hacerlo. No obstante, el primer paso es crear un servidor web.

En este capítulo, creará una VM Ubuntu Linux e instalará un servidor web básico. No se preocupe por tener que hacerla en Linux: al final del capítulo en el ejercicio de laboratorio

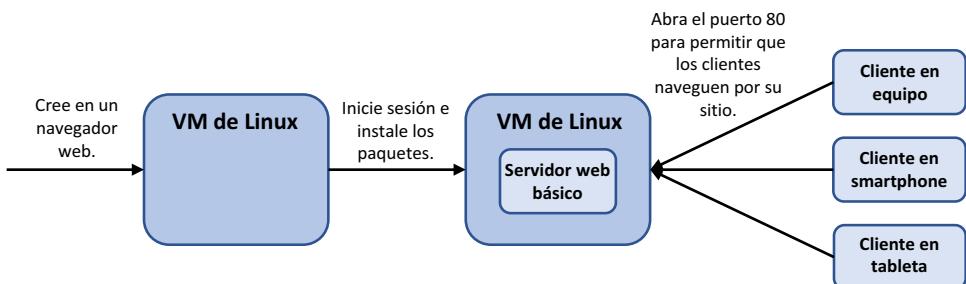


Figura 2.1 En este capítulo, creará una VM básica, iniciará sesión para instalar un servidor web y, a continuación, abrirá un puerto de red para que los clientes naveguen por el sitio web de ejemplo.

creará una VM Windows. Ubuntu es una plataforma de servidor web común y es una buena forma de aprender acerca de la autenticación de claves públicas SSH. A continuación, verá cómo abrir un puerto de red para que los clientes accedan a su sitio web en Internet. En la figura 2.1 se muestra una descripción general de este entorno básico.

2.1 Conceptos básicos de la configuración de la máquina virtual

Las máquinas virtuales son uno de los bloques de creación más comunes que utilizará cuando empiece a ejecutar aplicaciones en la nube. ¿Por qué? Suelen ser territorio conocido. La mayoría de los departamentos de TI ejecutan muchas cargas de trabajo utilizando Hyper-V o VMware en un entorno local, por lo que es probable que ya tenga cierta experiencia en la creación y ejecución de máquinas virtuales. Las organizaciones suelen dar sus primeros pasos en Azure con máquinas virtuales, ya que las cargas de trabajo IaaS no requieren el gran ajuste mental que hay que hacer cuando se empiezan a ejecutar cargas de trabajo PaaS.

Existen soluciones para migrar máquinas virtuales desde un entorno local como Hyper-V o VMware a Azure, pero antes de dejarse llevar por las posibilidades de Azure (algunas de las cuales exploraremos en capítulos posteriores), veamos algunos aspectos básicos. Estas páginas siguientes pueden parecer consideraciones y opciones conocidas que tiene con las máquinas virtuales locales. Si es así, ¡genial! Si esto es nuevo, no se preocupe; gran parte de la administración se obtiene en Azure, y cosas como las redes virtuales suelen crearse y configurarse una vez y luego se dejan tranquilas. En los próximos capítulos profundizaremos en cada área, así que respire hondo y vaya paso a paso.

2.1.1 Regiones y opciones de disponibilidad

Azure está dividido en regiones de todo el mundo, y cada región tiene uno o más centros de datos. Estos centros de datos entregan los recursos básicos de procesamiento, almacenamiento y red para ejecutar sus aplicaciones y cargas de trabajo. Azure se ejecuta en más de 50 regiones, y la lista crece cada pocos meses. Con tantas regiones, la idea es que pueda implementar aplicaciones cerca de sus empleados o clientes. Esta disponibilidad regional reduce la latencia y mejora la experiencia del usuario final.

Una región de Azure puede no ofrecer todos los servicios disponibles en Azure. Con cientos de servicios disponibles, el conjunto más común de servicios básicos suele estar en todas partes, pero los servicios nuevos o de nicho suelen aparecer con el tiempo. Mientras planifica sus aplicaciones en Azure, consulte la disponibilidad de los productos por región en <https://azure.microsoft.com/global-infrastructure/services>.

En el capítulo 8, veremos algunas de las opciones de alta disponibilidad, como los conjuntos de disponibilidad y las zonas de disponibilidad. Estas opciones de redundancia permiten a Azure distribuir varias instancias de sus máquinas virtuales o aplicaciones dentro de un único centro de datos o en toda una región. Esta capacidad le permite definir su tolerancia a las actualizaciones de mantenimiento o a los errores de hardware. En los primeros capítulos de este libro, normalmente solo creará una o dos máquinas virtuales, así que aún no se preocupe por estas opciones de disponibilidad.

2.1.2 Imágenes de VM

Para crear una máquina virtual (VM), necesita un punto de partida. Normalmente, este punto de partida se reduce a la elección del sistema operativo: Windows o Linux. Luego viene la elección de qué versión de Windows usar (como Windows Server 2016 o 2019) o qué distribución de Linux usar (como Ubuntu, Red Hat Enterprise Linux o SUSE).

Una *imagen* (un paquete de SO preconfigurado con opciones de configuración básicas aplicadas) es ese punto de partida. Azure contiene cientos de estas imágenes precompiladas en Azure Marketplace para utilizarlas al crear máquinas virtuales. Con frecuencia, se pueden aplicar las licencias de Windows existentes, dependiendo de su modelo de licencia actual, o se puede optar por el apoyo adicional de Canonical para ejecutar Ubuntu Linux o las actualizaciones de Red Hat, por ejemplo.

Para simplificar y acortar las cosas para que pueda completar estas lecciones en una hora de almuerzo, usará estas imágenes precompiladas en Azure a lo largo del libro. En el mundo real, probablemente querrá personalizar las cosas para adaptarlas a las necesidades y requisitos de su empresa. Para ello, a menudo creará sus propias imágenes de máquinas virtuales. El flujo de trabajo para crear y administrar la máquina virtual es el mismo que con las imágenes de Azure Marketplace, pero a menudo, crear sus propias imágenes requiere mucha planificación y luego horas de configuración, generalización y captura de sus propias imágenes por adelantado.

Pruébelo ahora

He aquí algunas ideas en las que debe pensar cuando planifique aplicaciones en Azure. Suelan básicas y, en muchas ocasiones, es posible que tome estas decisiones de forma automática, sin pensarlo mucho. Pero sigue siendo importante entender las necesidades de su aplicación antes de empezar a crear y ejecutar aplicaciones.

- ¿En qué regiones debe funcionar su aplicación? ¿Tiene una gran concentración de usuarios en una región específica? ¿Cómo ofrecerá la redundancia?

Si está desarrollando aplicaciones internas, ejecútelas en la región de Azure más cercana a nuestros usuarios. Por ejemplo, si tiene una oficina importante en Houston, Texas, (quizá le gusten los cohetes), ejecute sus aplicaciones y máquinas virtuales de Azure en el centro-sur de los Estados Unidos.

Si está desarrollando aplicaciones externas, ¿prevé tener clientes de determinadas regiones? Esta configuración puede requerir múltiples instancias implementadas en diferentes regiones (y también proporcionar alta disponibilidad). Llegaremos a esta configuración en el capítulo 12.

- ¿Necesita proporcionar muchas personalizaciones de VM? ¿Cuánto tiempo demora probar y validar todos esos cambios? ¿Qué necesidades empresariales los impulsan?

En un entorno tradicional local, se suele dedicar mucho tiempo a crear imágenes preconfiguradas para las implementaciones. En la nube, intente reducir este tiempo. Las imágenes precompiladas de Azure incluyen las actualizaciones de seguridad más recientes; se prueban para usted y luego se replican geográficamente para lograr los tiempos de implementación más rápidos.

Si crea sus propias imágenes, utilice características como Azure Shared Image Gallery para distribuir y replicar esas imágenes según se requiera (<https://docs.microsoft.com/azure/virtual-machines/windows/shared-image-galleries>).

2.1.3 Tamaños de las VM

Hay varias familias de tamaños de VM en Azure. Estas familias contienen grupos de tipos de hardware virtuales similares diseñados para ciertas cargas de trabajo. A veces, los tamaños se actualizan a medida que hay disponibles nuevas ofertas de hardware y cargas de trabajo, pero las familias principales siguen siendo las mismas. Los tipos de familia son los siguientes:

- *De uso general*: ideal para desarrollo y pruebas o para bases de datos y servidores web de producción de bajo uso.
- *Optimizado para proceso*: CPU de alto rendimiento, como para servidores de aplicaciones de producción.
- *Optimizado para memoria*: opciones de memoria más grandes, como cuando es necesario ejecutar grandes bases de datos o tareas que requieren un gran número de procesos de información in-memory.
- *Optimizado para almacenamiento*: rendimiento intensivo y de baja latencia del disco para aplicaciones con uso intensivo del disco.
- *GPU*: VM especializadas en gráficos basadas en NVIDIA, si necesita representación de gráficos o video.
- *Informática de alto rendimiento*: en pocas palabras ¡mucho de todo! Una gran cantidad de CPU, memoria y rendimiento de red para las cargas de trabajo más exigentes.

¿Qué tan grande puede ser la VM que cree en Azure? Las cosas mejoran constantemente, pero cuando se trata de crear la VM más grande, puede crear una VM serie Mv2 (parte de la familia optimizada para memoria) con 208 CPU virtuales y 5,7 TiB de memoria. Eso sería un servidor de Minecraft bastante decente, ¿no lo cree?

Lo importante a entender aquí es que el número de VM y la cantidad de CPU y memoria que puede solicitar en Azure solo están limitados por su presupuesto. Probablemente le sería muy difícil crear VM de este tamaño en el ámbito local tradicional.

Al crear una máquina virtual en Azure Portal o mediante la CLI o PowerShell, tiene que elegir qué tamaño de VM utilizar. Un tamaño común de VM, como D2s_v3, se usa a menudo como predeterminada para comenzar. Esto es probablemente demasiada potencia para un servidor web básico, pero es rápido para crear la VM e instalar los paquetes requeridos.

Azure Portal le permite filtrar en función de un tamaño aproximado (por ejemplo, pequeño, mediano o grande) o una familia específica (como las VM de uso general u optimizadas para memoria). También se muestra una estimación del costo mensual para que se haga una idea de lo que cuesta cada máquina virtual. Preste atención a los costos, ya que pueden acumularse rápidamente. Dentro de lo razonable, es posible administrar cambiar el tamaño de la VM después de que la VM esté en funcionamiento, aunque la máquina virtual tiene que apagarse y reiniciarse para completar el proceso.

Ahorrar costos de las VM

Las máquinas virtuales creadas por defecto suelen ser demasiado potentes para lo que necesita, pero son rápidas de implementar y utilizar, lo que ayuda a reducir la cantidad de tiempo que pasa instalando paquetes en su hora de almuerzo.

En el mundo real, preste atención a las demandas de memoria, CPU y almacenamiento de su máquina virtual. Cree VM del tamaño correcto. Este enfoque es el mismo que en el ámbito local, donde puede terminar con VM que tienen mucha más memoria o muchas más CPU virtuales asignadas de lo que necesitan.

También hay un tipo especial de VM en Azure: las serie *B*. Estos tamaños de VM utilizan recursos de memoria y CPU expandibles, además puede obtener créditos por procesos no utilizados. Si desea guardar sus créditos de Azure, puede elegir esta serie de VM para los ejercicios del libro. Viene con un punto de precio más bajo y son ideales para casos donde no siempre necesita mucha CPU y memoria. Pero tenga cuidado: dependiendo del tamaño de la VM de la serie B que cree, puede tener menos CPU y memoria que algo como la serie de D2s_v3, por lo que funcionará un poco más lento.

2.1.4 Azure Storage

El almacenamiento para VM en Azure es sencillo. ¿Cuántos discos desea, de qué tamaño y de qué tipo? Los dos primeros no son realmente específicos de Azure, así que los omitiremos. Estos tipos de almacenamiento están disponibles:

- *Discos SSD (unidad de estado sólido) premium*: utilice SSD de alto rendimiento y baja latencia que son perfectos para cargas de trabajo de producción. Debería utilizar principalmente este tipo para obtener el mejor rendimiento para sus aplicaciones.
- *Discos SSD estándar*: utilice SSD estándar y ofrezca un rendimiento coherente en comparación con las unidades de disco duro (HDD). Este tipo es ideal para cargas de trabajo de desarrollo y pruebas o para usos de producción de bajo presupuesto y demanda, como servidores web.
- *Discos HDD estándar*: utilice discos giratorios normales que son ideales para accesos menos frecuentes a los datos, como archivos de datos o copias de seguridad. No se recomienda este tipo para ejecutar cargas de trabajo de aplicaciones.

No es necesario profundizar mucho más en los detalles del almacenamiento para crear un servidor web rápido. Obtendrá más información en el capítulo 4, incluidos los discos Ultra que son solo para discos de datos conectados. Por ahora, es suficiente saber que cuando se elige un tamaño de VM, también ayuda a definir qué tipo de almacenamiento se requiere.

Los discos virtuales que utiliza, independientemente del tipo, se denominan *discos administrados de Azure*. Estos discos administrados le permiten crear una VM y conectar discos de datos adicionales sin preocuparse por las cuentas de almacenamiento subyacentes, los límites de recursos ni las asignaciones de rendimiento. Los discos administrados también se cifran automáticamente en reposo: ¡no necesita configurar nada para proteger sus datos! Nuevamente, en el capítulo 4 se aborda todo esto y mucho más. Por ahora, puede dejar que Azure cree el disco más adecuado según el tamaño de la máquina virtual que seleccione.

Pruébelo ahora

Para comprobar sus conocimientos, analice las siguientes preguntas:

- Para la mayoría de las cargas de trabajo de producción, ¿qué tipo de disco ofrece el mejor rendimiento?

Un disco SSD premium es normalmente lo que debería utilizar para las cargas de trabajo de producción. Este tipo suele ser la opción predeterminada cuando se crea una máquina virtual. Los discos SSD estándar son una segunda opción aceptable, y los ultra SSD solo deberían utilizarse en aplicaciones con uso intensivo del disco que requieran una baja latencia. Aunque hay un poco de ahorros de costos con los discos duros estándar (HDD), el rendimiento suele ser grande, al igual que en los entornos virtuales locales.

- ¿Qué familia de máquinas virtuales es una buena opción para un servidor de bases de datos?

Una VM optimizada para la memoria es una buena opción, ya que las bases de datos suelen necesitar una mayor cantidad de memoria que los recursos de CPU. Intente siempre estimar las necesidades de recursos y supervisar el rendimiento después de la implementación. No tenga miedo de cambiar el tamaño de la máquina virtual para obtener el rendimiento deseado.

2.1.5 Redes virtuales

Suena obvio, pero una VM necesita conectividad de red si desea que cualquier persona llegue a sus aplicaciones. Para un servidor web básico, se necesita tanto una red virtual como una conectividad externa. En el capítulo 5 se abordan en detalle las redes básicas de Azure, y el capítulo 9 se adentra en cómo distribuir el tráfico a múltiples máquinas virtuales mediante el uso de equilibradores de carga. Las cosas se ponen muy bien en el capítulo 11, con Azure DNS y el enrutamiento global de los usuarios finales con Traffic Manager. No lo convertiré en un ingeniero de redes, pero aprenderá mucho sobre las redes Azure en este libro.

Para empezar con los fundamentos necesarios para este capítulo, una red virtual en Azure se compone de las mismas características básicas que una red física normal:

- Un espacio de direcciones y una máscara de subred, como 10.0.0.0/16.
- Una o más subredes, que puede utilizar para dividir, por ejemplo, el tráfico externo, el de la base de datos o el de la aplicación.
- Tarjetas de interfaz de red (NIC) virtuales que conectan VM a una subred determinada.
- Direcciones IP virtuales que se asignan a recursos, como una NIC virtual o un equilibrador de carga.

Puede crear una VM que solo esté conectada a una red virtual sin proporcionar conectividad externa, como puede ser el caso de los servidores de aplicaciones o bases de datos de back-end. Para conectarse a estas VM para fines de administración y mantenimiento, puede crear una conexión de red privada virtual (VPN) o puede utilizar una conexión privada exclusiva con Azure desde su equipo de red local, denominada *ExpressRoute*.

El servidor web básico que desarrollará en este capítulo requiere un tipo específico de dirección IP virtual: una dirección IP pública. Esta dirección IP pública se asigna a la NIC virtual y permite que el tráfico externo llegue a su VM. Luego, puede controlar el flujo de tráfico a su VM con NSG (grupos de seguridad de red). Piense en un firewall normal que utilice para abrir o cerrar varios puertos y protocolos; en Azure, los grupos de seguridad de red bloquean el tráfico de forma predeterminada y solo permiten el tráfico específico que usted defina. El tráfico común para permitir sería HTTP o HTTPS en los puertos TCP 80 y 443. También se puede abrir la administración remota mediante el protocolo de escritorio remoto (RDP) o Secure Shell (SSH), con cuidado, lo que hará más adelante en este capítulo para ver cómo conectarte e instalar algunos paquetes.

2.2 Creación de un par de claves SSH para la autenticación

En el ejercicio de laboratorio del final del capítulo, creará lo que probablemente ya conoce: una VM de Windows Server. Este tipo de máquina virtual usa la autenticación basada en contraseña. Muchas aplicaciones en la nube funcionan con Linux; de hecho, más de la mitad de las máquinas virtuales en Azure funcionan con Linux. Normalmente no se utiliza la autenticación basada en contraseña con Linux; en su lugar, se utiliza SSH y un par de claves públicas. Para empezar a ampliar sus habilidades, el servidor web básico de este capítulo funciona con Linux, por lo que necesita sentirse cómodo con la creación y el uso de SSH. No *necesita* experiencia en Linux para trabajar en la nube, pero recomiendo encarecidamente que aprenda algunos de los fundamentos.

Pares de claves SSH

SSH es un protocolo utilizado para comunicarse de forma segura con equipos remotos y es la manera más común de iniciar sesión en las VM Linux. Es similar al uso de una conexión RDP a una máquina virtual de Windows, excepto que en Linux, toda la sesión SSH está típicamente basada en la consola. Con la criptografía de clave pública, puede utilizar un par de claves digitales para autenticarse con una VM de Linux remota.

Un par de claves SSH consta de dos partes: una clave pública y una clave privada. La clave pública se almacena en su VM Linux en Azure. Usted conserva una copia de la clave privada. Cuando inicie sesión en su VM Linux, la clave pública de la VM remota se compara con la clave privada que mantiene localmente. Si los pares de claves coinciden, se conecta a la VM. El proceso es algo más que eso, pero en esencia, la criptografía de clave pública es un medio fantástico para verificar la identidad.

Sería conveniente acostumbrarse a usar las claves SSH para iniciar sesión en las VM Linux, ya que son mucho más seguras que las contraseñas porque, entre otras cosas, no son susceptibles a ataques por fuerza bruta para obtener contraseñas. Debe centrarse siempre en la seguridad como concepto central, especialmente en la nube.

Pruébelo ahora

Crear un par de claves públicas SSH con Azure Cloud Shell:

- 1 Abra un navegador web y vaya a <https://portal.azure.com>. Inicie sesión en la cuenta de Azure que creó en el capítulo 1 y, a continuación, seleccione el ícono de Cloud Shell en la parte superior del panel, como se muestra en la figura 2.2. También puede abrir Cloud Shell directamente en <https://shell.azure.com>.



Figura 2.2 Seleccione el ícono de Shell para iniciar Cloud Shell en Azure Portal.

- 2 La primera vez que abra Cloud Shell, tomará unos minutos crear un almacenamiento permanente que luego estará siempre conectado a sus sesiones y le permitirá guardar y recuperar scripts, archivos de configuración y pares de claves SSH. Acepte las notificaciones para permitir la creación del almacenamiento.
- 3 Si es necesario, elija Bash en el menú desplegable de la esquina superior izquierda de Cloud Shell. También hay soporte para PowerShell, aunque nos centraremos principalmente en Bash y en la CLI de Azure a lo largo del libro.
- 4 Para crear un par de claves, ingrese el siguiente comando:

```
ssh-keygen
```

- 5 Acepte las indicaciones predeterminadas pulsando la tecla Intro. En un par de segundos, tendrá un par de claves públicas SSH que podrá usar con todas sus VM. El comando ssh-keygen tiene como valor predeterminado una clave de longitud de 2048 bits y utiliza el protocolo RSA versión 2. Este es un buen equilibrio de seguridad y se recomienda para la mayoría de los casos. La figura 2.3 muestra un ejemplo de un par de claves SSH completas en Cloud Shell.

A screenshot of an Azure Cloud Shell terminal window titled 'Bash'. The terminal displays the output of the 'ssh-keygen' command. It starts with 'Requesting a Cloud Shell... Succeeded.' followed by 'Connecting terminal...'. Then it says 'Welcome to Azure Cloud Shell' and provides instructions: 'Type "az" to use Azure CLI' and 'Type "help" to learn about Cloud Shell'. The user then runs 'ssh-keygen'. The terminal shows the generation of an RSA key pair. It asks for a file to save the key ('Enter file in which to save the key (/home/iain/.ssh/id_rsa)'). It creates a directory ('Created directory '/home/iain/.ssh''). It asks for a passphrase ('Enter passphrase (empty for no passphrase)'). It then asks for a second passphrase ('Enter same passphrase again:'). It confirms that the identification has been saved ('Your identification has been saved in /home/iain/.ssh/id_rsa'). It also saves the public key ('Your public key has been saved in /home/iain/.ssh/id_rsa.pub'). It shows the key's fingerprint ('The key's randomart image is:'). Finally, it shows the SHA256 key fingerprint ('-----[SHA256]-----'). The session ends with 'iain@Azure:~\$ []'.

```
Bash
Requesting a Cloud Shell... Succeeded.
Connecting terminal...
Welcome to Azure Cloud Shell

Type "az" to use Azure CLI
Type "help" to learn about Cloud Shell

iain@Azure:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/iain/.ssh/id_rsa):
Created directory '/home/iain/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/iain/.ssh/id_rsa.
Your public key has been saved in /home/iain/.ssh/id_rsa.pub.
The key's randomart image is:
-----[RSA 2048]-----
|   .+o.|
| o +oo |
| B. ==o |
| .+.+o..|
| + .S ...B+..|
| + ..oo X.0+ |
| E . oo.= B..|
| o. o .|
| ...
-----[SHA256]-----+
iain@Azure:~$ [ ]
```

Figura 2.3 Un par de claves SSH creado en Azure Cloud Shell con el comando ssh-keygen

- 6 Para ver la clave pública y usarla con una VM, escriba el siguiente comando:

```
cat .ssh/id_rsa.pub
```

- 7 Seleccione la salida y cópiela en un archivo de texto simple en su equipo. Utilizará esta clave pública para crear una VM en la sección 2.3; se hace referencia a esta VM desde la CLI de Azure en el resto del libro. Normalmente, no es necesario copiar y pegar toda la clave cada vez, pero es bueno ver lo que ocurre al principio. Esta información no es supersecreta, por lo que utilizar el Bloc de notas oTextEdit para crear y guardar una copia de la clave está bien. Tenga cuidado al copiar el resultado de la clave pública, ya que es sensible a espacios en blanco adicionales o caracteres faltantes. Un ejemplo de una clave pública SSH completa es la siguiente:

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDPGaOBsfhJJJOHAWAv+RLLR/vdUTzS9HOIj
➥ JyzWWLsnu0ESH2M6R+YYPPNXv9X7dmVyM1zCXXeLucpnyFjevbwPedxTgifyxgCFTgyrlr1
➥ kg7o4EyCTGBGhTA+hSHuhXGXa12KPdKWehsPwHMa6Hs8fbt/in9Z1k2ZAwbvBt+LWPcmJgNO
➥ FuolIHOSOEeoQQqdXLrGa7NU/3fzSXdt9Y2BT1KLInc4KnwdOuONddLw3iANvK+Gkwax8iK
➥ 7IickMoammwwJUCRF+MTEK9pZ84tfsc9qOIAdhrCCLbQhtoWjZpIwYnFk+SNBE8bZZtB8b2
➥ vkDFNZ1A5jcAd6pUR3tPuL0D iain@cc-a444-9fdee8b2-2014310619-v5c15
```

CONSEJO Cloud Shell se basa en el navegador, así que los métodos abreviados de teclado para copiar y pegar pueden ser un poco diferentes de lo que está acostumbrado. Debe utilizar Ctrl-Insert y Mayús-Insert para copiar y pegar, en lugar de Ctrl-C y Ctrl-V.

2.3 Creación de una VM desde su navegador web

Ahora que conoce un poco de la teoría de las VM de Azure y ha creado un par de claves SSH, está listo para lanzarse a crear una máquina virtual. Voy a ponerlo en marcha y luego dejaré que configure la VM basándose en lo que acaba de aprender, así que definitivamente tendrá que prestar atención.

Las herramientas de CLI de Azure y Azure PowerShell son increíblemente potentes, pero una gran fortaleza de Azure es cuánto tiempo ha dedicado a crear una gran experiencia en el portal. Azure Portal es una herramienta gráfica en la web que le permite ver cómo se unen todos los diferentes componentes y hacer una comprobación visual rápida para asegurarse de que todo esté bien. El portal incluye un par de cosas únicas que las otras herramientas no proporcionan, además es rápido de usar porque no necesita instalar nada.

Pruébelo ahora

La creación de una VM en Azure le da un montón de valores predeterminados que puede usar para reducir el número de opciones que tiene que hacer. Para este ejercicio, mire los recursos que Azure va a crear basándose en lo que ha aprendido en la sección 2.2 para cosas como la red y el almacenamiento:

- 1 En Azure Portal (<https://portal.azure.com>), seleccione Crear un recurso en la esquina superior izquierda del panel. Los recursos más populares deberían aparecer en la lista, incluyendo la versión más reciente de Ubuntu con soporte a largo plazo (LTS) (en el momento de escribir este artículo, se trata de Ubuntu Server 18.04 LTS).

2 Seleccione la versión LTS.

También puede buscar en Marketplace en la parte superior de la ventana o navegar por la lista de servicios de alto nivel (como informática y redes) para tener una idea de qué más está disponible para sus propias necesidades futuras. Intente continuar con Ubuntu Server 18.04 LTS para poder seguir uno de los próximos ejercicios para instalar los componentes del servidor web.

3 Cree un grupo de recursos para su servidor web.

Cuando se crean recursos en Azure, estos están lógicamente contenidos en un grupo de recursos que usted define. Estos grupos suelen contener recursos afines para sus aplicaciones. El capítulo 7 aborda las formas de planificar y administrar las aplicaciones mediante el uso de grupos de recursos.

Por ahora, le sugerimos nombrar los grupos de recursos por capítulos para organizar las cosas a medida que avanza. Por ejemplo, coloque el nombre azuremol - chapter2, para el grupo de recursos de este ejercicio.

4 Dele a su VM un nombre, como webvm, y luego elija una región cercana a usted.
Por ahora, no se preocupe por la redundancia de la infraestructura.

Mira las opciones para la imagen de la VM, solo para tener una idea de las otras opciones, pero para este ejercicio, quedese con Ubuntu Server 18.04 LTS. El tamaño de la VM predeterminado está bien para este ejercicio, pero de nuevo, mire para ver lo que está disponible y cómo se consulta para los diferentes tamaños y qué hardware se ejecuta. Vea cómo los tamaños se alinean con las familias de máquinas virtuales que vio con anterioridad en este capítulo.

5 Asegúrese de que está utilizando la autenticación de clave pública SSH y luego proporcione un nombre de usuario, como azuremol. Utilizará este nombre de usuario para acceder a la máquina virtual en el siguiente ejercicio.

6 Copie y pegue la clave pública SSH que creó en la sección anterior. Una vez más, asegúrese de que no hay espacios en blanco ni formatos adicionales cuando copie y pegue la clave pública. La clave SSH tiene que estar en una línea. Incluso el ajuste de palabra en el Bloc de notas puede provocar problemas. Azure Portal valida la clave antes de que pueda continuar.

7 Para conectarse a la VM en el siguiente ejercicio e instalar los componentes de servidor web, abra SSH en el puerto 22.

Abrir SSH en una máquina virtual pública no es una gran práctica de seguridad. En el capítulo 16 se estudia cómo abrir y restringir el acceso de forma automática utilizando el acceso de VM Just-in-Time.

Observe algunos de los otros puertos que puede abrir aquí. HTTP y HTTPS son puertos comunes para abrir, y se supone que está creando un servidor web en este capítulo, ¿verdad? No haga trampas y abra esos puertos todavía; quiero presentarle la CLI de Azure en el siguiente ejercicio, donde permitirá el tráfico HTTP.

Conéctese de forma segura mediante un host bastión

En escenarios reales, no debe abrir los puertos de administración remota para SSH o RDP a la Internet pública. En serio, ¡no lo haga! Siga los procedimientos recomendados que debe usar en el mundo físico local, como conectarse únicamente cuando sea necesario y limitar el acceso remoto a un conjunto específico de direcciones de administración.

(continuación)

Una manera común de ofrecer acceso remoto es utilizar un host bastión, o un servidor de salto. En este tipo de configuración, no se conecta directamente a los servidores de aplicaciones desde su equipo portátil o de escritorio. En su lugar, se conecta a un host bastión dedicado y, luego, se conecta al servidor que necesita administrar. Este enfoque bloquea el acceso a un conjunto limitado de direcciones y brinda una forma segura de permitir la administración remota.

Azure Bastion (<https://docs.microsoft.com/azure/bastion>) ofrece un enfoque administrado para esta necesidad de conexión remota segura. Cree un host de Azure Bastion en una subred dedicada y, a continuación, utilice este host para conectarse a las máquinas virtuales que ejecutan sus aplicaciones. No es necesario que estas sean accesibles públicamente. Puede hacer todo a través de Azure Portal sin abrir puertos de red para SSH o RDP. El propio host bastión se administra para usted en términos de actualizaciones de seguridad y reglas de grupo de seguridad de la red.

- 8 Observe algunas de las otras opciones de configuración de la VM para el almacenamiento y la red para familiarizarse con las opciones, aunque puede dejar todos los valores predeterminados por ahora.
- 9 También hay algunas opciones de administración interesantes, como activar el apagado automático, las copias de seguridad y los diagnósticos, que se tratan en los capítulos 12 y 13. Por ahora, desactive cosas como los diagnósticos de arranque y los diagnósticos de invitados del SO, ya que tiene que crear y configurar una cuenta de almacenamiento para que funcionen.
- 10 Cuando esté listo, revise y cree su máquina virtual básica.

2.4 Conexión a la VM e instalación del servidor web

Cuando su VM esté en funcionamiento, puede utilizar la clave SSH que creó antes para iniciar sesión en la VM. Luego, puede comenzar a instalar y configurar el servidor web, y puede hacerlo todo a través del Cloud Shell.

2.4.1 Conexión a la VM con SSH

En esta sección se examina cómo puede obtener rápidamente los detalles de conexión para su VM.

Pruébelo ahora

Si Linux es nuevo para usted, ¡no se preocupe! Siga los próximos pasos para iniciar sesión en su VM:

- 1 En Azure Portal, busque y seleccione Máquinas virtuales en la barra de navegación del lado izquierdo de la pantalla. La creación de la VM a partir del ejercicio anterior tarda un par de minutos, así que seleccione el botón Actualizar hasta que el estado de la VM muestre *En ejecución*. Cuando esté listo, elija su máquina virtual y seleccione Conectar, como se muestra en la figura 2.4.

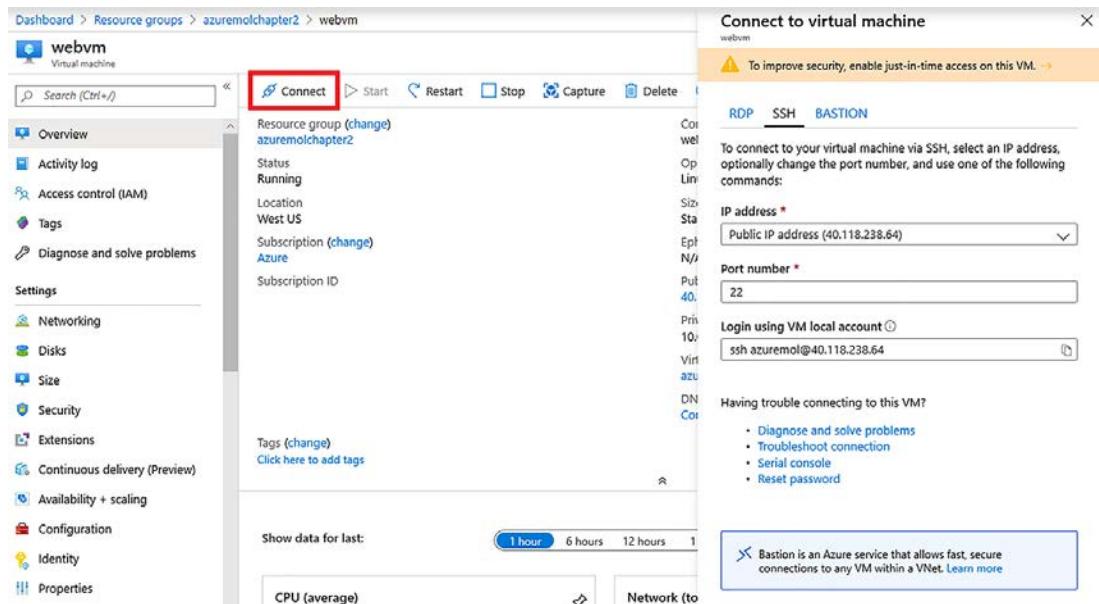


Figura 2.4 Seleccione su VM en Azure Portal y, a continuación, seleccione Conectar para generar la información de conexión de SSH.

Con una VM Linux, se le mostrará el comando SSH que incluye su nombre y la dirección IP pública. Copie este comando de conexión SSH, como `ssh azuremol@104.209.208.158`.

En una VM Windows, el botón Conectar descarga en el equipo un archivo de conexión RDP donde viene incluida la dirección IP pública de la VM.

- 2 Si es necesario, abra Cloud Shell otra vez. Si va a cambiar entre Cloud Shell y el portal, puede minimizar Cloud Shell para mantenerlo disponible en segundo plano.
- 3 Pegue el comando SSH en Cloud Shell y, a continuación, presione Intro. La clave SSH creada anteriormente se utiliza automáticamente para autenticación.

La primera vez que se conecta a una VM con SSH, le solicita que la agregue a una lista de hosts de confianza. Este es otro nivel de seguridad que ofrece SSH. Si alguien intenta interceptar el tráfico y dirigirlo a una VM remota diferente, su cliente SSH local sabe que algo ha cambiado y le avisa antes de conectarse.

Acepte la opción para guardar la conexión de VM remota. La figura 2.5 muestra el proceso de conexión SSH en Azure Cloud Shell.

```

Bash    | ? {} >
ia in@Azure:~$ ssh azurem01@104.209.208.158
The authenticity of host '104.209.208.158 (104.209.208.158)' can't be established.
ECDSA key fingerprint is SHA256:fg8PUA2A9g1ID0z4gZluZfZ9uXNSTVLKbqNm5UYSW5w.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '104.209.208.158' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 5.0.0-1014-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

 System information as of Wed Aug 21 03:24:32 UTC 2019

 System load: 0.26      Processes:          130
 Usage of /:   4.2% of 28.90GB  Users logged in:   0
 Memory usage: 4%        IP address for eth0: 10.0.1.4
 Swap usage:  0%

 0 packages can be updated.
 0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

azurem01@webvm:~$ 

```

Figura 2.5 Utilice la cadena de conexión que se muestra en Azure Portal para crear una conexión SSH a su VM desde Cloud Shell.

En este punto, hay dos opciones: está lejos de casa o el aviso de Linux es totalmente ajeno. No se preocupe. No necesita saber una gran cantidad de comandos de Linux, y cada comando se explicará a medida que avanzamos. Dicho esto, le recomendamos encarecidamente que aprenda al menos algunas habilidades básicas de administración de Linux. Gran parte de la nube se basa en los sistemas Linux, y estamos presenciando un gran cambio hacia los contenedores y microservicios para la compilación y la administración de aplicaciones. Si es un administrador de Windows de la vieja escuela, es igualmente bienvenido. Hay algo preparado para usted al final del capítulo, así que tenga paciencia.

2.4.2 Instalación del servidor web

¿Crear una VM? Verificado. ¿Se conectó a la VM con SSH? Verificado. Ahora puede instalar los paquetes para un servidor web y prepararse para verlo en acción.

Azure es compatible con muchas distribuciones de Linux (*distros*). Las herramientas de administración de paquetes y las ubicaciones de archivos de configuración varían un poco entre distros. Vá a usar Ubuntu en este libro porque es una de las distribuciones de Linux más populares y bien documentadas para informática en la nube. Si se queda atascado en el camino, de seguro podrá encontrar un montón de documentación que lo ayudará, comenzando por <https://help.ubuntu.com>. Si desea utilizar una distribución diferente con la que ya se sienta cómodo, siéntase libre de usarla. De lo contrario, quédese con Ubuntu.

Pruébelo ahora

Desde su sesión SSH a la VM, instale los paquetes de servidor web con APT:

- 1 En Ubuntu, los paquetes se instalan con Advanced Packing Tool (APT), una herramienta de administración de paquetes muy potente que instala de manera automática cualquier paquete adicional que necesite. Todo lo que tiene que hacer es decir "Instalar un servidor web", y la APT instala todos los componentes necesarios.

Para este ejemplo, instale la pila web LAMP. Este es probablemente el conjunto más común de componentes web: Linux, Apache (un servidor web), MySQL (un servidor de base de datos) y PHP (un lenguaje de programación web):

```
sudo apt update && sudo apt install -y lamp-server^
```

El primer comando actualiza los paquetes disponibles, lo cual es una buena práctica para asegurarse de instalar los mejores y más recientes paquetes. Cuando el comando termina, se ejecuta el siguiente comando con `&&`. ¿Por qué no empezar una nueva línea para el siguiente comando? `&&` ejecuta el siguiente comando solo si el comando anterior se completó correctamente. Si, por ejemplo, no había conectividad de red para que apt obtuviese los paquetes más recientes (obviamente, hay que tener conectividad de red para conectarse en primer lugar), no tiene sentido ejecutar el comando `install`.

Si el comando `update` tiene éxito, luego `apt` determina qué paquetes adicionales necesita y comienza a instalar `lamp-server`. ¿Por qué hay un símbolo de separación al final (^)? El símbolo le dice a `apt` que instale todo el conjunto de paquetes que componen el servidor LAMP y no un solo paquete denominado `lamp-server`.

- 2 El instalador puede solicitarle una contraseña, o usar una contraseña de MySQL vacía por defecto. Eso no es muy seguro, y para un uso real de la producción, es necesario especificar una contraseña segura. En el capítulo 15, las cosas se ponen más interesantes y se guarda una contraseña fuerte y segura en Azure Key Vault que se incorpora automáticamente a este asistente de instalación de MySQL.

Se tarda un minuto más o menos en instalar todos los paquetes de la pila web LAMP, y entonces habrá terminado.

- 3 Escriba `exit` para cerrar la sesión de su VM y volver al mensaje de Cloud Shell.

¡Listo! Su servidor web está funcionando, pero aún no podrá acceder a él desde un navegador web. Para ello, es necesario permitir que el tráfico web llegue a la VM.

2.5 Permitir que el tráfico web llegue a la VM

El servidor web está funcionando, pero si ingresa la dirección IP pública de su VM en un navegador web, la página web no se cargará. ¿Por qué? ¿Recuerda los grupos de seguridad de la red de los que se habló brevemente en la sección 2.1.5? Cuando creó la VM, se creó un grupo de seguridad de red para usted y se agregó una regla que permite la administración remota: en este caso, la regla era SSH. El resto de la VM está bloqueada. Para permitir que los visitantes accedan a su servidor web a través de Internet, es necesario crear una regla en el grupo de seguridad de red que permita tráfico web. De lo contrario, ¡nadie puede pedir pizzas!

2.5.1 Creación de una regla para permitir el tráfico web

En esta sección se mezclan un poco las cosas utilizando la CLI de Azure para crear una regla para el tráfico web. Podría haber abierto este puerto HTTP en el portal cuando creó la máquina virtual, pero entonces se habría perdido la mitad de la diversión.

La CLI de Azure está disponible en Cloud Shell. No hay nada que necesite instalar. En el capítulo 5 se abordan las redes virtuales y los grupos de seguridad de red con más profundidad; por ahora, puede comprobar lo rápido y potente que es la CLI de Azure con un solo comando.

Pruébelo ahora

Abra Azure Cloud Shell y siga estos pasos para ver la CLI de Azure en acción:

- 1 Si cerró la ventana de Cloud Shell, ábrala de nuevo desde Azure Portal. Asegúrese de que se cargue el shell de Bash, y no de PowerShell. Si es necesario, cámbielo a la versión Bash.
- 2 Para ver la CLI de Azure y los módulos instalados, escriba az --version. Se muestra una lista de módulos y números de versión. Lo genial sobre Cloud Shell es que siempre tiene la versión más reciente y mejor disponible.

NOTA Si es observador, puede haber notado que la información de salida del comando es sobre la versión de Python. ¿Por qué es importante esta información? Python es un lenguaje de programación poderoso y popular. La CLI de Azure está escrita en Python, que es en parte el motivo que lo hace multiplataforma y disponible para que lo instale localmente en cualquier equipo si no quiere utilizar siempre Cloud Shell. Para mantenerse al tanto de la iniciativa de Microsoft por contribuir a la comunidad open source, la CLI de Azure está disponible en GitHub para que cualquier persona realice contribuciones, sugerencias o reporte problemas (<https://github.com/Azure/azure-cli>).

- 3 Para abrir un puerto, especifique el nombre de la VM y su grupo de recursos, junto con el número de puerto. Para tráfico web, debe abrir el puerto 80. Escriba el grupo de recursos (-g) y el nombre de la VM (-n) que especificó cuando la creó:

```
az vm open-port -g azuremolchapter2 -n webvm --port 80
```

2.5.2 Visualización del servidor web en acción

Ahora que tiene un puerto abierto en su VM, veamos qué pasa cuando intenta acceder a él desde un navegador web:

- 1 En Azure Portal, seleccione su VM si navegó lejos de este. La dirección IP pública aparece en la esquina superior derecha de la página de descripción general de la máquina virtual.
- 2 Seleccione la dirección y cópiela.
- 3 En su navegador web, abra una nueva pestaña o ventana y pegue en la dirección IP pública. Se cargará el sitio web predeterminado de Apache, como se muestra en la figura 2.6. Todavía no parece una pizzería, pero tiene las bases listas para traer su código y empezar a compilar su aplicación.

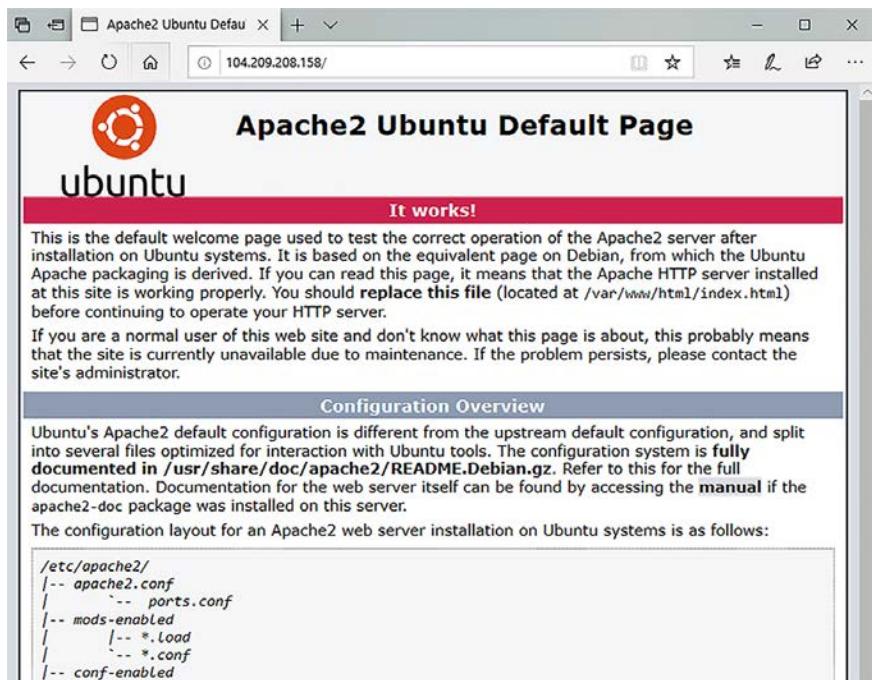


Figura 2.6 Para ver el servidor web en acción y ver la página predeterminada de Apache 2, ingrese la dirección IP pública en un navegador web.

2.6 Laboratorio: Creación de una VM Windows

En las secciones anteriores repasamos paso a paso cómo instalar la pila LAMP en una VM Linux Ubuntu. Esta plataforma es común para los sitios web, pero es posible que deba dedicarle un poco más de tiempo si solo tiene conocimientos de Windows. Los equipos de desarrollo o quienes toman las decisiones comerciales podrían querer utilizar .NET, por ejemplo. Incluso así, puede ejecutar .NET Core en las VM Linux, así que no tome la decisión en función del lenguaje.

Tomando lo que aprendió en el ejemplo paso a paso, intente crear una VM que ejecute Internet Information Services (IIS). Estas son algunas sugerencias:

- Necesita una VM que ejecute Windows Server 2019.
- Usted usa RDP, no SSH, así que espere una experiencia de conexión ligeramente diferente.
- En Server Manager, busque una opción para agregar roles y funciones.
- Necesita instalar el servidor web (IIS).
- No olvide abrir un puerto de red para el tráfico HTTP en el puerto TCP 80. Puede utilizar el portal si lo desea.

2.7 Limpieza de recursos

A medida que crea recursos en Azure, los gastos comienzan a andar. Se le cobra por minuto, así que es prudente formar buenos hábitos y no dejar recursos como VM ejecutándose cuando haya terminado con ellas. Tiene dos maneras de detener los cargos de facturación por ejecutar una VM:

- *Desasignar una VM.* Puede seleccionar el botón Detener del portal para detener y desasignar una VM, que libera todos los procesos de red e informáticos mantenidos.
- *Eliminar una VM.* Esta opción es bastante obvia. Si no queda nada en Azure, no hay nada que pagar. Asegúrese de haber terminado con la VM antes de eliminarla. ¡No existe el botón Deshacer en Azure!

Le recomendamos que cree un grupo de recursos para cada implementación de la aplicación a medida que comience a compilar en Azure. Eso es lo que hará a medida que avance por los ejercicios de este libro. Si nombra los grupos de recursos por capítulos, como azuremolchapter2, será más fácil realizar un seguimiento de sus recursos y saber qué eliminar. Esta práctica permite que la limpieza sea un poco más fácil, ya que puede eliminar todo el grupo de recursos al final de cada capítulo. Elija Grupos de recursos en el menú de navegación a la izquierda de la pantalla, abra cada grupo de recursos que haya creado en este capítulo y, a continuación, seleccione Eliminar grupo de recursos, como se muestra en la figura 2.7. Para confirmar, se le solicitará el nombre del grupo de recursos.

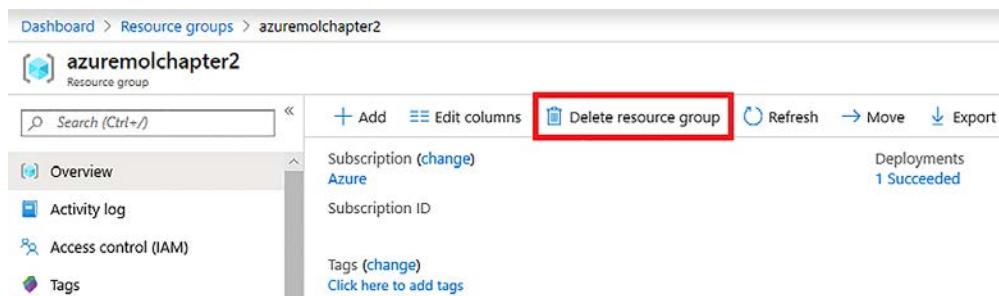


Figura 2.7 Para ahorrar costos, elimine los grupos de recursos cuando ya no los necesite.

Si tiene el hábito de eliminar recursos cuando ha terminado con ellos, podrá completar cómodamente este libro con los créditos gratuitos de Azure. Al finalizar todo, desasigne su VM al final de cada lección para que pueda reanudar la actividad el día siguiente y detener los costos de facturación.

2.8 Houston, tenemos un problema

En ocasiones se topará con problemas en Azure. Hay que decirlo. Por lo general, la plataforma Azure es buena en relación con los problemas que surgen a medida que crea recursos:

- Los informes de la CLI de Azure o Azure PowerShell se van respaldando a medida que ejecuta los comandos, por lo que debiera ser obvio cuando algo va mal. Azure PowerShell normalmente utiliza un agradable y tranquilo texto rojo para llamar su atención.
- La CLI de Azure puede ser un poco más críptica, ya que generalmente incluye las respuestas reales a las llamadas subyacentes de API REST desde el servidor. Si esto es nuevo, puede necesitar algunos éxitos y fracasos para entender lo que está haciendo mal. La parte útil de obtener las respuestas de REST es que puede copiar y pegar los mensajes de error en su motor de búsqueda favorito y, por lo general, obtendrá resultados concretos que lo ayudarán a solucionar los problemas.

¿Necesita un descanso? ¡Acabamos de empezar!

Al abrir una página web en su navegador, el equipo se comunica con un servidor web mediante HTTP. Casi puedo garantizar que ya ha visto un mensaje de error 404 en un sitio web. Ese mensaje significa que no se pudo encontrar la página web. Otros errores comunes que puede haber visto son 403 (no tiene permiso para ver la página) y 500 (el servidor encontró un error).

Incluso cuando las cosas van bien, podría no darse cuenta de que su navegador recibe mensajes de código 200 cuando la página se carga bien, o mensajes de código 301 si una página se ha redirigido a una nueva ubicación. No necesita entender ni hacer un seguimiento de todos estos códigos; son solo unas maneras estándar en que HTTP hace posible la comunicación entre los equipos.

Anteriormente, en este capítulo se habló de cómo crear y administrar los recursos de Azure a través del portal web, la CLI o PowerShell. Se accede a todos los servicios de Azure a través de interfaces de programación de aplicaciones de transferencia de estado representacional (REST) (API).

Si esto es nuevo para usted, las API REST son algo así como una forma estandarizada de exponer los servicios vía HTTP. Se utilizan solicitudes HTTP estándar, como GET y POST para solicitar información o realizar un cambio, y cuando la plataforma acepte y procese la solicitud, recibirá un mensaje del estado. Azure tiene un conjunto bien definido de API REST.

No necesita entender lo que significa todo esto. Solo tenga en cuenta que cuando vea un mensaje de error, no siempre estará en el formato más útil y fácil de leer para los humanos. A veces, se obtiene la respuesta HTTP bruta de la API REST que debe descifrar usted mismo. De nuevo, pegue este error en su motor de búsqueda favorito. Es muy probable que alguien se haya encontrado con el problema y haya proporcionado una razón más legible para lo que salió mal y lo que hay que corregir.

Los problemas más comunes con las VM se producen cuando se conecta a su VM. Puede conectarse para administración remota con SSH o RDP, o intentar acceder a sus aplicaciones a través de un navegador web o cliente de escritorio. Por lo general, estos

problemas suelen estar relacionados con la red. No vamos a culpar totalmente a la gente de las redes hasta el capítulo 5, así que aquí hay un par de cosas que debe comprobar:

- ¿Se puede conectar a otras VM o aplicaciones de Azure que funcionen en Azure? Si no puede, algo local de su red está probablemente impidiendo el acceso.

Si puede conectarse a otros recursos de Azure, asegúrese de haber abierto las reglas de grupo de seguridad de red (sección 2.5). En el capítulo 5 se profundiza en estas reglas.

- Para problemas de autenticación, pruebe lo siguiente:
 - Confirme que tiene las claves SSH correctas. Azure debería decirle en el momento de crear la VM si la clave pública no es válida, pero si tiene más de una clave privada, asegúrese de utilizar la correcta.
 - Para problemas RDP, intente conectarse a localhost\<username> e ingrese su contraseña. De forma predeterminada, la mayoría de los clientes RDP intentan presentar credenciales locales o credenciales de red que su VM no entenderá.

Azure Web Apps

En el capítulo 2, creó una máquina virtual (VM) e instaló paquetes manualmente para ejecutar un servidor web básico. Si estuviera ansioso por empezar, podría compilar una pizzería en línea con esta VM. Uno de los mayores casos de uso de Azure VM es para ejecutar aplicaciones web, normalmente a escala. Las aplicaciones web son una carga de trabajo cómoda para las VM. La comodidad es buena, si también le gusta el mantenimiento que conlleva la administración de todas esas máquinas virtuales. Ya sabe, cosas divertidas como las actualizaciones de software, los parches de seguridad, el registro centralizado y los informes de cumplimiento. ¿Qué pasaría si pudiera obtener toda la potencia de un servidor web seguro para ejecutar sus aplicaciones web, incluida la capacidad de escalado automático para satisfacer las demandas, pero sin la necesidad de crear y administrar todas esas VM? Permítame presentarles el servicio Azure Web Apps.

En este capítulo, vamos a comparar el enfoque de infraestructura como servicio (IaaS) de las VM y los servidores web con el enfoque de plataforma como servicio (PaaS). Aprenderá las ventajas de Azure Web Apps a medida que crea una aplicación web y verá cómo trabajar con sus versiones de desarrollo y producción. A continuación, aprenderá a implementar su aplicación web automáticamente desde un control de código fuente, como GitHub. Este flujo de trabajo se muestra en la figura 3.1. Azure Web Apps le permite implementar y ejecutar su pizzería.

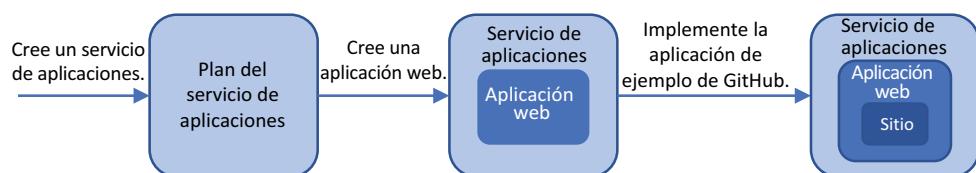


Figura 3.1 En este capítulo, creará un plan de App Services y una aplicación web básica, y luego implementará un sitio web desde GitHub.

en línea en cosa de minutos, sin necesidad de instalar y configurar un VM ni paquetes de servidor web.

3.1 Información general y conceptos de Azure Web Apps

Con Azure Web Apps, empezará a sumergir sus dedos en el maravilloso mundo de las soluciones PaaS. Si piensa que la informática en la nube solo se trata de VM, probablemente cambiará de idea. Al comienzo de este libro, hablamos sobre adquirir potencia para el equipo y centrarse en sus aplicaciones y clientes. Cuando se cambia de soluciones IaaS, como VM y se vuelca hacia soluciones PaaS, como aplicaciones web, sus aplicaciones y clientes se convierten en el foco.

Para ejecutar aplicaciones web en IaaS, las VM requieren la administración del SO, actualizaciones de aplicaciones, reglas de seguridad y tráfico, además de la configuración de todo el sistema. Con Web Apps, usted carga su aplicación web y delega todas esas tareas administrativas. Ahora puede enfocarse en mejorar la experiencia de la aplicación para sus clientes o en mejorar la disponibilidad con opciones de escalado y administración de tráfico.

¿Significa eso que nunca debe ejecutar VM para hospedar una aplicación web? Probablemente no. Hay razones válidas para ejecutar toda la pila de aplicaciones y configurarla usted mismo, como por ejemplo si necesita un soporte de aplicaciones muy específico o un tiempo de ejecución de lenguaje. Pero Web Apps puede proporcionar muchos de los casos de uso para ejecutar aplicaciones web.

3.1.1 Lenguajes y entornos compatibles

¿Qué tipo de flexibilidad ofrece Web Apps en términos de lenguajes de programación que puede utilizar para crear su aplicación web? ¡Bastante! Existen dos plataformas principales para ejecutar Web Apps: Windows y Linux. Puede ejecutar aplicaciones web .NET Core, Node.js, Python, Java, Ruby y PHP de forma nativa en instancias de aplicaciones web de Windows y Linux. En Windows, también puede ejecutar .NET Framework completo. Si quiere ser realmente especial y ejecutar su aplicación web en contenedores, también existe Web Apps for Containers que le permite ejecutar contenedores Docker nativos para Linux. Nos adentraremos más en contenedores y Docker en el capítulo 19: Por ahora, es suficiente que entienda que sus opciones están cubiertas con Web Apps.

¿En qué situaciones Web Apps no tiene mucho sentido? No todos los lenguajes de aplicación son compatibles con Web Apps. Digamos que realmente quiere torturarse con una aplicación web escrita en Perl. En ese caso, es probable que vuelva a ejecutar en VM IaaS que administra usted mismo, porque Perl no es compatible con Web Apps. Sin embargo, Web Apps es compatible con los lenguajes de programación web más comunes que podría querer utilizar. Probablemente también debería buscar una versión más reciente de la aplicación que una escrita en Perl.

Web Apps no solo es compatible con varios lenguajes, sino que también con varias versiones de esos lenguajes. Piense en PHP, por ejemplo. Típicamente, hay tres o cuatro versiones de PHP que puede seleccionar para tener una mejor compatibilidad para su aplicación. Y lo mejor de todo, es que no tiene que preocuparse por las dependencias en el servidor web subyacente para que sean compatibles con todo, como lo necesitaría si usted mismo administrara una VM IaaS. Python es otro ejemplo de diferencias entre las versiones estables 2.7 y 3.6 (y posteriores), como se muestra en la figura 3.2.

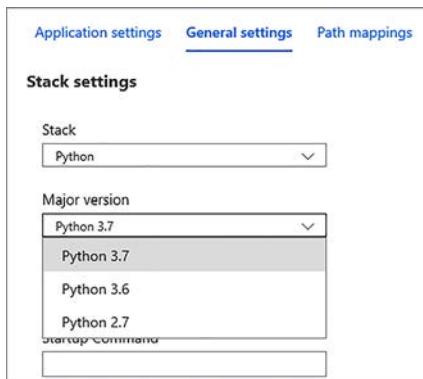


Figura 3.2 Seleccione una versión específica de un lenguaje en la configuración de la aplicación de Web Apps.

Web Apps también mantiene actualizadas las correcciones de seguridad. Pero no espere que una versión anterior de PHP o Python siga siendo compatible indefinidamente, en algún momento habrá un corte de versiones anteriores compatibles. Una vez más, ese podría ser un momento en el que deba volver a ejecutar usted mismo VM IaaS si la aplicación necesita una versión anterior de lenguaje. Pero si necesita ejecutar una versión anterior de un lenguaje determinado para que sea compatible con una aplicación heredada, no se quede estancado en un enfoque de mantenimiento constante. Siempre busque cambiar esas aplicaciones heredadas a plataformas compatibles más modernas.

3.1.2 Representación de versiones diferentes mediante ranuras de implementación

Las ranuras de implementación proporcionan un entorno preconfigurado para su aplicación web. Puede insertar nuevas versiones de la aplicación en una ranura de implementación y ejecutarlas con variables de entorno o conexiones a base de datos, sin afectar al sitio en vivo. Cuando esté satisfecho con lo que ve en la ranura de implementación, puede cambiar esta versión al sitio en vivo en un instante. A continuación, el sitio que antes estaba en vivo luego cambia a su propia ranura de implementación, proporcionando una versión archivada o puede llevar la aplicación de vuelta a producción si fuera necesario.

El número de ranuras de implementación disponibles varía en función del nivel de aplicación web que seleccione. Un mayor número de ranuras de implementación permite que diferentes desarrolladores utilicen varias versiones en etapas, a medida que representan y prueban sus propias actualizaciones.

3.1.3 Planes de App Services

Web Apps es parte de la familia más extensa App Services de Azure, que también incluye Mobile Apps, API Apps y Logic Apps. Todas, salvo Logic Apps están disponibles en todas las regiones donde Azure está disponible. Este es un excelente recurso para comprobar la disponibilidad por región: <https://azure.microsoft.com/regions/services>. Muchos servicios están disponibles en todo el mundo.

Cuando necesite crear un recurso de App Services, como una aplicación web, cree o utilice un plan de App Services existente. El plan de servicio define la cantidad de recursos disponibles para usted, cuánta automatización está disponible para escalar y hacer copias

de seguridad de su aplicación web, y cuánta disponibilidad tiene su sitio con espacios de ensayo y Traffic Manager (una forma de dirigir geográficamente el tráfico a la instancia más cercana para un usuario, que veremos en el capítulo 11). Así como con cualquier cosa, obtiene lo que paga. Las necesidades de su aplicación y de su negocio deben guiarle en cuanto a la cantidad de recursos requeridos y las características adicionales que se necesitan. Cada nivel de servicio se basa en las funciones de los niveles inferiores, agregando generalmente más almacenamiento y recursos disponibles.

Los cuatro niveles principales del plan de servicios son los siguientes:

- *Gratis/compartido*: utiliza una infraestructura compartida, ofrece almacenamiento mínimo y no tiene opciones para implementar versiones diferentes en etapas, enrutar tráfico o hacer copias de respaldo. El nivel compartido le permite utilizar un dominio personalizado y cobra por este dominio.
- *Básico*: proporciona recursos informáticos dedicados para su aplicación web y le permite utilizar SSL y escalar manualmente el número de instancias de la aplicación web que ejecuta. Los niveles gratuitos/compartidos y básicos ofrecen un buen entorno para probar el servicio de Web Apps, pero no recomendaría ejecutar ninguna carga de trabajo real de producción o desarrollo. El rendimiento no es un factor limitante, pero se echan de menos algunas de las funciones automatizadas, como las copias de seguridad y el escalado.
- *Estándar*: agrega copias de respaldo diarias, escalado automático de instancias de aplicaciones web y ranuras de implementación, y permite enrutar usuarios con Traffic Manager. Este nivel puede ser adecuado para aplicaciones de baja demanda o entornos de desarrollo en los que no se necesita un gran número de copias de seguridad o ranuras de implementación.
- *Premium*: proporciona copias de respaldo más frecuentes, mayor capacidad de almacenamiento y un mayor número de ranuras de implementación y opciones de escalado de instancias. Este nivel es ideal para la mayor parte de las cargas de trabajo de producción.

Caso para aislamiento

Con las soluciones PaaS como Web Apps, la infraestructura se abstrae intencionalmente. Como lo indican los nombres de algunos de los niveles de plan de servicio, las aplicaciones web se ejecutan a través de una plataforma compartida de recursos disponibles. Eso no quiere decir para nada que las aplicaciones web no son seguras y que otros pueden ver sus datos privados. Pero el cumplimiento de normas o razones reglamentarias pueden requerir que ejecute sus aplicaciones en un entorno aislado y controlado. Escriba *entornos de App Services*: entornos aislados que le permiten ejecutar instancias de App Services como Web Apps en una parte segmentada de un centro de datos de Azure. Usted controla el tráfico de red entrante y saliente y puede implementar firewalls y crear conexiones de red privada virtual (VPN) en sus recursos locales.

Todos estos componentes de infraestructura todavía se abstraen en gran medida con los entornos de App Services, pero este enfoque proporciona un gran equilibrio cuando se busca la flexibilidad de las soluciones PaaS, pero a su vez desea conservar algunos controles más específicos del flujo de tráfico de las conexiones de red.

Puede hacer mucho con los niveles Gratuito y Básico, aunque para cargas de trabajo de producción probablemente debería utilizar el nivel Estándar o Premium. El ejemplo de este capítulo utiliza el nivel Estándar para que pueda ver todas las funciones disponibles. Cuando utiliza Web Apps con sus propias aplicaciones, puede decidir cuántas de estas funciones necesita y seleccionar el nivel de plan de App Services más adecuado.

3.2 Creación de una aplicación web

Con algo de teoría bajo la manga, echemos un vistazo a una Web App en acción. Hay un par de pasos para poner en marcha una aplicación. En primer lugar, usted crea la aplicación web básica y visualiza el sitio predeterminado en el navegador. Luego, usa la página web de GitHub de ejemplo y la inserta en Azure. Tal vez sus desarrolladores web han empezado a compilar un front-end para su pizzería en línea, así que tiene un sitio web básico listo para cargar.

NOTA Si nunca ha usado Git antes, no se preocupe. No necesita entender lo que Git está haciendo en este punto, y tendrá la oportunidad al final del capítulo para jugar y explorar un poco. *Aprenda Git en un mes de almuerzos*, de Rick Umali (<https://www.manning.com/books/learn-git-in-a-month-of-lunches>), es una excelente introducción al uso de Git si quieres aprender un poco más, y está disponible para leer de forma gratuita en la plataforma liveBook de Manning.

3.2.1 Creación de una aplicación web básica

Al igual que en el capítulo 2, le voy a dar algunas pautas a lo largo del camino, pero observe si puede aplicar algo de la teoría sobre los tiempos de ejecución de la aplicación y los planes de App Service para crear una aplicación web. Si no está seguro de algunas opciones, es seguro aceptar los valores predeterminados por ahora.

PaaS, no IaaS

Esta aplicación web es un nuevo recurso y está separada de las VM como la que creó en el capítulo 2, que es un enfoque de IaaS para crear y ejecutar aplicaciones web. El enfoque de PaaS es Web Apps. No hay una relación real entre los dos tipos. De hecho, si siguió el consejo en el capítulo 2 y eliminó su VM, esta aplicación web se ejecuta sin necesidad alguna de una VM en su suscripción de Azure.

Pruébelo ahora

Complete los siguientes pasos para crear su aplicación web:

- 1 Abra un navegador web y vaya a <https://portal.azure.com> inicie sesión en su cuenta de Azure.
- 2 En el portal, seleccione Crear un recurso en la esquina superior izquierda del panel.
- 3 Seleccione Web en la lista de recursos que puede crear y, a continuación, seleccione Web App.
- 4 Para ayudar a mantener las cosas limpias y organizadas como hizo en el capítulo 2, le sugiero que cree un grupo de recursos dedicado para su aplicación web, como azuremolchapter3.

- 5 Para el nombre de la Web App, escriba un nombre único globalmente. Este nombre debe ser globalmente único, ya que crea la URL a su aplicación web en el formato `http://<name>.azurewebsite.net`. En el caso que se lo esté preguntando, sí: aquí puede colocar un nombre de dominio personalizado. Por ahora, utilice la dirección `azurewebsites.net` predeterminada.
- 6 Va a insertar un código HTML básico, no un contenedor Docker, pero busque en todas las diferentes pilas de tiempo de ejecución disponibles. Puede cambiar esta configuración después de haber creado la aplicación web, pero por ahora, elija un tiempo de ejecución ASP.NET que se ejecute en Windows.
- 7 Permita que Azure cree un plan de App Service de manera automática, pero cambie el tamaño a estándar S1. Este nivel proporciona todas las funciones básicas sin proporcionar demasiados recursos para su sitio web de demostración básico. En las implementaciones del mundo real, este paso es donde podría crear y configurar manualmente sus propios planes de App Service o seleccionar un plan existente.
- 8 Cuando esté listo, revise y cree su primera aplicación web.

App Services tarda unos segundos en crearse. Cuando esté listo, busque y seleccione App Services en la barra de navegación de la izquierda de la pantalla; a continuación, elija su aplicación web de la lista. En la ventana Información general de su aplicación web, vea y seleccione la URL de la aplicación web, como `https://azuremol.azurewebsites.net`.

Al seleccionar la URL de la aplicación web, se abrirá una nueva ventana o pestaña del navegador. Se carga la página de la aplicación web predeterminada, como se muestra en la figura 3.3. Todavía no parece una pizzería...

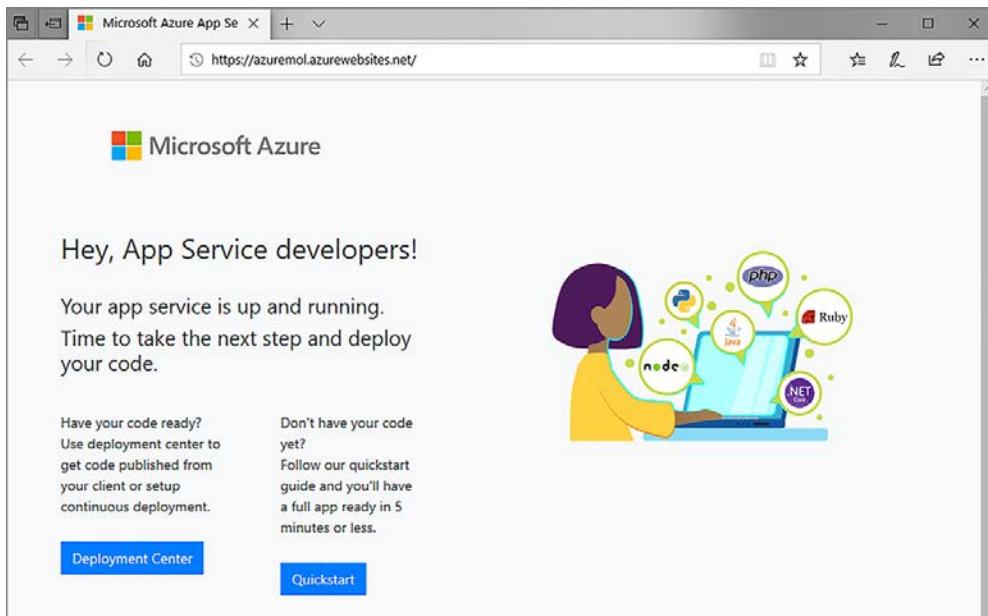


Figura 3.3 Para ver la página de aplicación web predeterminada en acción, abra un navegador web e ingrese la URL de su sitio web.

3.2.2 Implementación de un sitio HTML de ejemplo

Tiene una aplicación web en Azure, pero es un sitio web aburrido y predeterminado. ¿Cómo hacer su propio sitio web en Azure? Una de las formas más comunes de hacerlo entre plataformas es usando Git. La mayoría de los desarrolladores y equipos de aplicaciones utilizan un sistema de control de código fuente. En lugar de almacenar archivos en el equipo y guardar los cambios a medida que va avanzando, los sistemas de control de código fuente mantienen un seguimiento de los cambios y le permiten trabajar con otros. Puede crear versiones de prueba que no afecten su código de producción y volver a versiones anteriores si surgen problemas. Git es uno de los sistemas de control de código fuente más comunes; GitHub es un servicio basado en la nube que le permite compartir y contribuir código con el resto del mundo. Microsoft adquirió GitHub en junio de 2018, pero no hay nada que lo obligue a usar GitHub con Azure, o viceversa. Todos los ejemplos de este libro están disponibles en GitHub.

Para este ejemplo, usted crea una copia local del sitio HTML estático de ejemplo y, a continuación, inserta los archivos a la aplicación web de Azure. Este flujo de trabajo se muestra en la figura 3.4.



Figura 3.4 Va a crear una copia local de los archivos de ejemplo de GitHub con el comando `git clone`. Para insertar estos archivos locales a la aplicación web de Azure, utilice `git push`.

Pruébelo ahora

Complete los siguientes pasos para obtener una copia de la página HTML de ejemplo de GitHub e insertarla a su aplicación web:

- 1 Abra Cloud Shell en Azure Portal y espere unos segundos para que su sesión se conecte. Para comenzar, necesita el sitio HTML de ejemplo de GitHub.

Para clonar o copiar el sitio HTML de ejemplo de GitHub, ingrese el siguiente comando:

```
git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
```

Si esta es su primera vez con Git en Cloud Shell, es necesario definir un par de ajustes para que Git entienda quién es usted. Para la mayoría de los ejercicios en este libro, realmente no importa; pero para utilizar con sus propios proyectos y aplicaciones, es ideal para hacer un seguimiento de quién realiza ciertas acciones en un sistema de control de código fuente. Solo tiene que definir esta configuración una vez. Escriba su propia dirección de correo electrónico y nombre completo en `git config` de la siguiente manera:

```
git config --global user.email "iain@azuremol.com"
git config --global user.name "Iain Foulds"
```

- 2 Cambie al directorio azure-mol-samples-2nd-ed que se creó al clonar el repositorio Git:

```
cd azure-mol-samples-2nd-ed/03/prod
```

- 3 A fin de estar preparado para cargar la página HTML de ejemplo, inicialice Git y luego agregue y confirme sus archivos. No se preocupe demasiado por los comandos Git en este momento. Debe decirle a Git qué archivos rastrear y agregar, y busque la forma de poder hacer un seguimiento de los cambios más tarde si fuera necesario:

```
git init && git add . && git commit -m "Pizza"
```

- 4 Ahora puede prepararse para insertar este sitio HTML de ejemplo en su aplicación web. Primero, defina las credenciales de implementación. Para proteger Web Apps cuando usa un método de implementación como Git o FTP, tiene que proporcionar un nombre de usuario y una contraseña. Las Web Apps pueden utilizar un conjunto de credenciales válidas en todos los planes de App Service en Azure o credenciales de nivel de aplicación que solo se aplican a una sola aplicación.

En el mundo real, recomiendo utilizar credenciales a nivel de aplicación para minimizar el alcance de un ataque en caso de que una de las credenciales quede expuesta. Azure genera automáticamente las credenciales a nivel de aplicación, pero tiene que recuperar y asignar estas credenciales cada vez. Para simplificar las cosas, utilice una credencial definida que pueda reutilizar en los próximos capítulos.

Cree las credenciales de despliegue y especifique su propio nombre de usuario y contraseña segura. El nombre de usuario tiene que ser único globalmente. Si ayuda, agregue sus iniciales al nombre de usuario o una convención de nombres que tenga sentido para su entorno.

```
az webapp deployment user set --user-name azuremol --password @azuremol!
```

- 5 A continuación, necesita obtener la URL del repositorio Git de su aplicación web. Escriba el nombre de la aplicación web (no el nombre de usuario que creó en el paso 4) y el grupo de recursos que especificó cuando se creó la aplicación web para ver la URL del repositorio Git.

La barra invertida

En el siguiente ejemplo y los capítulos posteriores, la barra invertida (\) significa que el comando continúa en la línea siguiente. Es una forma de juntar largas líneas, y este enfoque se utiliza en una gran cantidad de ejemplos en línea donde puede copiar y pegar los comandos. No tiene que escribir las barras invertidas en los ejemplos de este libro si no quiere. Simplemente siga tipeando los parámetros adicionales como parte de una línea grande.

Si está utilizando el sistema de notificaciones de comandos de Windows en lugar de un Bash Shell, no incluya las barras invertidas. Si lo hace, no obtendrá para nada el resultado que desea.

```
az webapp deployment source config-local-git \
--name azuremol \
--resource-group azuremolchapter3 -o tsv
```

- 6 Su aplicación web está configurada para trabajar con repositorios Git, así que debe decirle a Cloud Shell qué repositorio es. En Git, usted define estas ubicaciones como *remotas*.

Copie su URL de Git clone del paso 5 y, a continuación, defina esta dirección URL como destino para el sitio HTML de ejemplo en Cloud Shell con el siguiente comando:

```
git remote add azure your-git-clone-url
```

- 7 Para cargar o copiar archivos con Git, los *debe* insertar. ¿A dónde los inserta Git? A un *remoto* como el que configuró en el paso 6, tal como *azure*. La parte final del comando es una rama, típicamente *maestra*. Gracias a las ramas en Git es como podrá hacer un seguimiento de los diferentes modelos de trabajo en curso. Una práctica recomendada en los entornos de producción es insertar para liberar ramas que puede nombrar como desee, tal como *dev* o *staging*. Estas ramas adicionales permiten que su código de producción funcione normalmente; a continuación, puede trabajar en nuevas funciones o actualizaciones de forma segura y sin afectar las cargas de trabajo reales que utilizan sus clientes.

Inserte el sitio HTML de ejemplo en su aplicación web:

```
git push azure master
```

- 8 Cuando se le solicite, escriba la contraseña que creó para las credenciales de implementación. Puede copiar y pegar la contraseña para minimizar los errores aquí.

En el resultado puede ver que se elimina la página de sitio web de la aplicación predeterminada existente y que el sitio HTML de ejemplo se carga y configura para ejecutarse. Este es un resultado de ejemplo:

```
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 510 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
remote: Updating branch 'master'. remote: Updating submodules.
remote: Preparing deployment for commit id 'ddaa01e9d86'.
remote: Generating deployment script.
remote: Generating deployment script for Web Site
remote: Running deployment command...
remote: Handling Basic Web Site deployment.
remote: Creating app_offline.htm
remote: KuduSync.NET from: 'D:\home\site\repository' to:
'D:\home\site\wwwroot'
remote: Deleting file: 'hostingstart.html'
remote: Copying file: 'index.html'
remote: Deleting app_offline.htm
remote: Finished successfully.
remote: Running post deployment command(s)...
remote: Deployment successful.
To https://azuremolikf@azuremol.scm.azurewebsites.net/azuremol.git
 * [new branch]      master -> master
```

Para ver la aplicación web actualizada, actualice su sitio en un navegador web o ábralo de nuevo desde la ventana de Información general en Azure Portal. Debería verse como el maravilloso ejemplo de la figura 3.5. Sí, el sitio es básico, pero el flujo de trabajo para implementar el sitio HTML estático más básico en una aplicación web compleja de .NET o Node.js, ¡es el mismo!

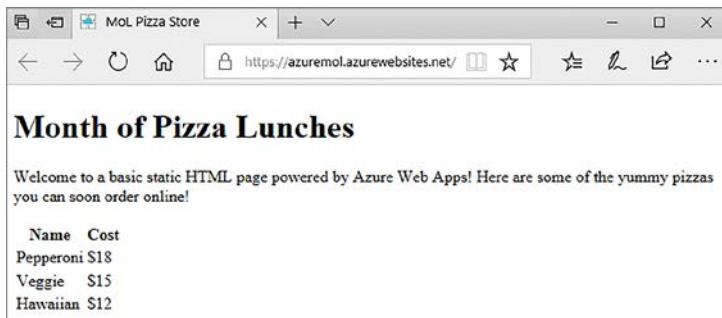


Figura 3.5 Actualice su navegador web para ver la página Web App predeterminada sustituida por el sitio HTML estático básico de GitHub.

3.3 Visualización de registros de diagnóstico

Ahora que ha visto cómo crear una aplicación web básica e implementar un sitio HTML simple, ¿qué pasa con la administración general? Si tiene problemas, sería útil ver el servidor web o los registros de la aplicación. Para ayudar a solucionar los problemas de sus aplicaciones, puede escribir el resultado de la aplicación en estos archivos de registro, que se pueden ver en tiempo real o escribirse en archivos de registro para revisión posterior.

Su aplicación web se ejecuta en gran medida por sí sola. No hay mucho que pueda hacer desde la perspectiva de mantenimiento en el host web subyacente. Si la aplicación tiene problemas, le recomendamos que mire los registros para ver lo que está pasando y solucionar el asunto. Con Azure Web Apps, puede configurar cosas como el nivel de los mensajes de registro a revisar, dónde almacenar los registros, y cuánto tiempo mantener los registros. La figura 3.6 describe cómo se generan y se ven los archivos de registro con Web Apps.

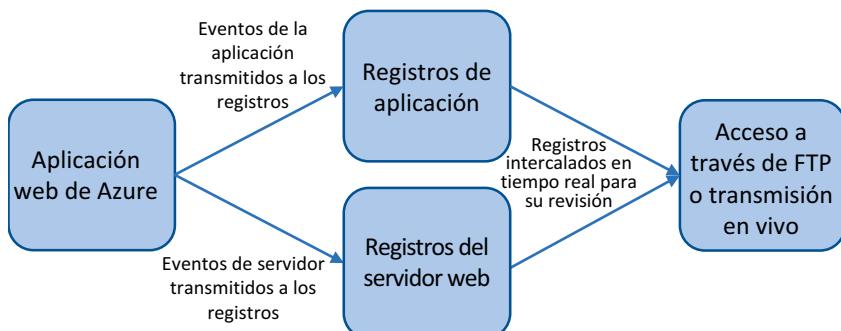


Figura 3.6 La aplicación puede generar registros de aplicación y registros de servidor, los que se pueden descargar mediante FTP o verse en tiempo real para ayudarlo a revisar o solucionar problemas.

Pruébelo ahora

Complete los siguientes pasos para configurar los registros de diagnóstico para su aplicación web:

- 1 En Azure Portal, seleccione la aplicación web que creó en el ejercicio anterior.
- 2 En la ventana Información general, desplácese hacia abajo hasta la sección Monitor y seleccione Registros de App Service.
- 3 Revise las opciones de registro disponibles, como el contenido y si desea habilitar el seguimiento de solicitudes fallidas. Si se encarga de la infraestructura de Azure, podría tener que trabajar con los desarrolladores de aplicaciones a fin de determinar los registros que necesitan para ayudar a solucionar problemas. A continuación, puede activar el registro correspondiente aquí. Los registros pueden almacenarse en el sistema de archivos local de la aplicación web o enviarse a Azure Storage para su procesamiento con otra aplicación.
- 4 Por ahora, active el registro de aplicaciones (sistema de archivos). También active el registro del servidor web en el sistema de archivos con un periodo de retención de siete días. El nivel de error predeterminado puede no mostrar nada si todo funciona bien, pero tenga cuidado al cambiar a Depurar o Seguimiento, ya que sus registros pueden llenarse con rapidez y hacer que sea difícil verrealmente lo que está sucediendo. Si tiene un problema, aumente gradualmente el nivel de registro hasta que capture suficiente información para solucionar los problemas sin verse abrumado por los datos de registro.

Si de verdad quiere profundizar en los datos, puede acceder a los registros almacenados en el sistema de archivos mediante FTP. Las direcciones FTP se muestran en la sección Registros de descarga o en la ventana Información general de la aplicación web. Puede estar pensando: "FTP es una forma complicada de obtener registros de diagnóstico. ¿No hay una manera más fácil?" ¡Porque sí la hay! En Azure Portal, justo donde configuró sus registros de diagnóstico, hay una opción de Transmisión de registros. ¿Puede adivinar lo que hace? Déjeme darle una pista: tiene que ver con la transmisión de sus archivos de registro.

Si selecciona este botón en Azure Portal, puede elegir entre Registros de aplicaciones y Registros de servidor web. Estos registros leen de los mismos registros de diagnóstico que se escriben en el archivo. Puede demorar algunos minutos para que aparezcan los datos de registro en la transmisión, y lo que se muestra depende de los niveles de registro que especifique y si la aplicación web genera algún evento de aplicación. Para el sitio HTML básico, la transmisión es bastante aburrida, pero es una función fantástica para tener en el navegador web. La figura 3.7 muestra registros de transmisión de servidor web de ejemplo en Azure Portal.

Pruébelo ahora

Vea los archivos de registro de transmisión en Azure Portal. Es posible que deba actualizar la página en su navegador web un par de veces para generar actividad en los registros.

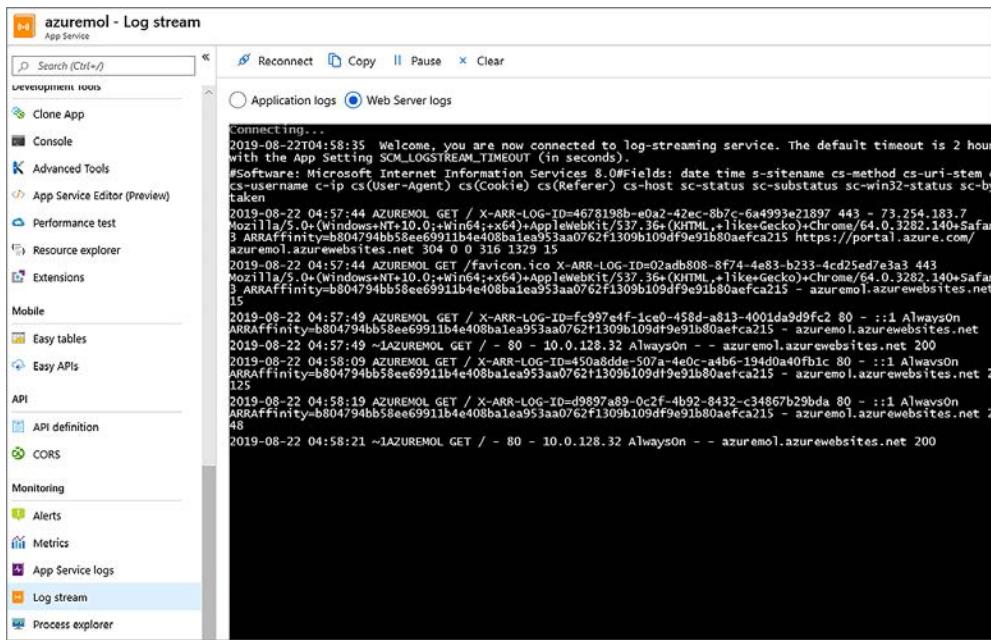


Figura 3.7 Puede ver las transmisiones de registro de servidores web de Web Apps correspondientes a registros en vivo desde la aplicación para poder verificar y depurar el rendimiento de la aplicación. La caja de la consola en el lado derecho de la pantalla muestra los registros de transmisión en tiempo real de su aplicación web.

A medida que se vaya sintiendo más cómodo con Azure y utilice el módulo CLI de Azure o Azure PowerShell, podrá transmitir registros con estas herramientas. Los desarrolladores también pueden habilitar la depuración remota con Visual Studio o configurar Application Insights para permitir que su aplicación web proporcione telemetría a servicios adicionales para supervisión y diagnóstico. La conclusión clave aquí es que a medida que se mueve hacia soluciones PaaS como Web Apps, todavía puede obtener registros de diagnósticos y datos de las aplicaciones cruciales para solucionar problemas y supervisar el rendimiento de su aplicación web.

3.4 Laboratorio: Creación y utilización de una ranura de implementación

Ya vio cómo crear un sitio HTML simple e insertar la página a Azure Web Apps con Git. ¿Qué pasa si ahora desea agregar algunos nuevos estilos de pizzas y verlos antes de poner el sitio en vivo para que los clientes los comiencen a pedir? Revise cómo utilizar una ranura de implementación para proporcionar un lugar para cargar sus cambios, revisarlos y luego llevarlos a producción:

- 1 En la aplicación web, seleccione Ranuras de implementación. Agregue una ranura de implementación llamada Dev, pero no clone ninguna configuración de la ranura de implementación existente.
- 2 Cuando esté listo, seleccione el espacio de ensayo de la lista. El portal muestra las mismas opciones de configuración y de registro que la ranura de producción, lo

que muestra cómo se puede cambiar la configuración en esta ranura de implementación sin afectar al sitio en vivo.

- 3 En esta ocasión, explore las opciones en Azure Portal para el Centro de implementación. Desea utilizar Git local para el control de código fuente que usa el servicio de compilación de App Service para el espacio de ensayo. Esto ocurrió en segundo plano cuando utilizó la CLI de Azure en el ejercicio anterior, pero dispone de otras opciones en términos de desde dónde puede implementar su código y qué servicio desarrolla y crea esa implementación.
- 4 Cuando haya terminado, copie Git Clone Uri, como <https://azuremol-dev.scm.azurewebsites.net:443/azuremol.git>. Observe cómo el repositorio Git incluye -dev para el espacio de ensayo.

Se incluye un sitio de desarrollo de ejemplo en los ejemplos de GitHub que clonó anteriormente.

- 5 En Azure Cloud Shell, cámbiese al directorio de desarrollo de la siguiente manera:

```
cd ~/azure-mol-samples-2nd-ed/03/dev
```

- 6 Como lo hizo anteriormente, inicialice, agregue y confirme los cambios en Git con los siguientes comandos:

```
git init && git add . && git commit -m "Pizza"
```

- 7 Cree un vínculo al nuevo repositorio de Git en el espacio de ensayo con git remote add dev, seguido de su URL de implementación del espacio de ensayo de Git.
- 8 Utilice git push dev master para insertar los cambios en la ranura de implementación.
- 9 Seleccione la dirección URL que dirige a su espacio de ensayo desde la ventana Información general de Azure Portal. El cambio no es muy grande, seguro, pero el título de la página le permite saber que está viendo la versión de desarrollo.
- 10 En Azure Portal para la aplicación web, ¿qué cree que ocurre si selecciona el botón Intercambiar, como se muestra en la figura 3.8? Pruébelo y, a continuación, actualice la página principal, como <https://azure-mol.azurewebsites.net>, en su navegador web.

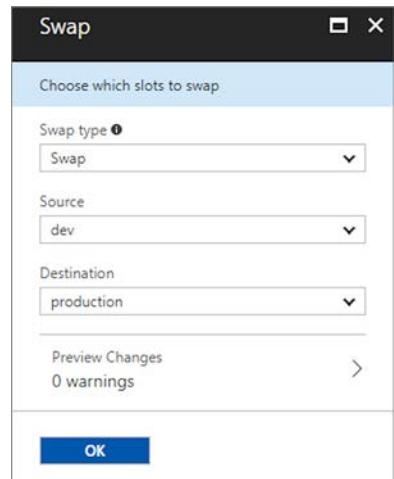


Figura 3.8 Cuando se intercambian ranuras para una aplicación web, se eligen las instancias de origen y destino que se van a cambiar. También puede previsualizar el nuevo aspecto antes de realizar los cambios en directo.

Ranuras de implementación, detrás de las escenas

Al intercambiar ranuras, lo que estaba en vivo en la ranura *production* ahora está en la ranura *dev* y lo que estaba en *dev* ahora está en vivo en *production*. No todos los ajustes se pueden intercambiar, como la configuración SSL y los dominios personalizados; pero en su gran mayoría, las ranuras de implementación son una gran manera de organizar y validar el contenido antes de dejarlo en vivo para sus clientes. También puede realizar un intercambio con *vista preliminar*, lo que le da la oportunidad de garantizar que el contenido intercambiado funcione de manera correcta antes de que esté públicamente en vivo en producción.

Para usar en producción en los flujos de trabajo DevOps, también puede configurar Intercambio automático. Aquí, cuando se observa una confirmación de código en el control de código fuente como GitHub, puede desencadenar una compilación en una ranura de implementación de Azure Web Apps. Cuando la compilación está completa y la aplicación está lista para servir el contenido, las ranuras de implementación se intercambian automáticamente para que el código se torne en vivo en producción. Usted suele usar este flujo de trabajo con un entorno de prueba para revisar primero los cambios de código, no para publicar en vivo directamente a producción.

Introducción a Azure Storage

Podemos estar seguros de una cosa en el mundo de la TI: cuando las cosas están mal, podemos culpar inevitablemente al almacenamiento y a las redes. Y lo digo porque fui administrador de red de área de almacenamiento en una vida pasada. El equipo de redes era mi mejor amigo. Estoy bromeando (acerca de ser mejores amigos), pero no importa lo bien que una aplicación esté diseñada y escrita: las piezas de la infraestructura fundacional deben ser sólidas para que funcione adecuadamente. Quédese conmigo en el próximo par de capítulos mientras exploro Azure Storage y Azure Networking. Puede que esté tentado a conocer a grandes rasgos estos servicios para hacer lo que es más entretenido en los capítulos posteriores, pero pasar algo de tiempo explorando y aprendiendo estos servicios básicos puede resultar muy valioso. No hará que su comida tenga un mejor sabor, pero puede ayudar a sus clientes mientras esperan la entrega de su deliciosa pizza.

En este capítulo se analizan los diferentes tipos de almacenamiento en Azure y cuándo utilizarlos. También hablo de las opciones de redundancia y replicación del servicio Azure Storage, y de cómo obtener el mejor rendimiento para sus aplicaciones.

4.1 Discos administrados

Hace años, el almacenamiento en servidores era costoso, lento y demasiado complicado. No era raro que un proveedor de almacenamiento le vendiera hardware que costaba cientos de miles de dólares y tardara días, o incluso semanas, para que un ejército de consultores e ingenieros lo configurara. A medida que la virtualización comenzó a arraigarse en los centros de datos, y VMware y Hyper-V se volvieron más aceptados, el almacenamiento se convirtió en el cuello de botella. Y qué hablar de las diferencias de firmware entre los adaptadores de almacenamiento en el servidor y la matriz de almacenamiento, las rutas de red redundantes que fallan de ida y vuelta, y el hecho que los discos de estado sólido (SSD) se consideren la única forma de ganar rendimiento.

¿Acaso Azure ha arreglado mágicamente todos estos problemas de almacenamiento? ¡Por supuesto que no! Pero se encarga del 95 % de estas preocupaciones y lo deja concentrarse en cómo compilar y crear experiencias increíbles para sus clientes. En este capítulo se aborda el último 5 % que debe considerar.

El servicio Azure Managed Disks simplifica el enfoque del almacenamiento de máquinas virtuales. Los discos administrados eliminan gran parte del trabajo que se realiza en segundo plano para ofrecerle... bueno, un disco. Eso es todo lo que debe preocuparse por las máquinas virtuales: el tamaño y la velocidad que tienen, y a qué se conectan. A lo largo del libro, y en todas sus implementaciones en el mundo real, siempre debe utilizar discos administrados para las máquinas virtuales. Los discos administrados son la opción predeterminada, y no hay muchas buenas razones para cambiar ese comportamiento.

Antes de los discos administrados, había que crear una cuenta de almacenamiento con un nombre único, limitar el número de discos virtuales que se creaban en cada uno y mover manualmente las imágenes de disco personalizadas para crear VM en distintas regiones. Estos tipos de discos se conocen como *discos no administrados* o *discos clásicos*. El servicio de discos administrados elimina la necesidad de una cuenta de almacenamiento, lo limita a "solo" 50 000 discos por suscripción y le permite crear VM de una imagen personalizada en todas las regiones. También obtiene la capacidad de crear y utilizar copias instantáneas de discos, cifrar automáticamente los datos en reposo y utilizar discos de hasta 64 TiB de tamaño.

¿Por qué es tan importante? Si se encuentra con documentación antigua o publicaciones en blog, podría tener que crear una cuenta de almacenamiento para sus VM. ¡Deténgase ahí mismo! Sí, puede convertir VM de discos no administrados a discos administrados, pero si parte desde cero, inicie cada proyecto con discos administrados desde el principio. El caso de uso de los discos no administrados se utiliza más para mantener la compatibilidad con versiones anteriores de las implementaciones existentes, aunque yo argumentaría que debería convertir esas cargas de trabajo en discos administrados.

4.1.1 Discos del SO

¿Recuerda que mencionamos que si quería obtener un mejor rendimiento, debía comprar SSD? No hay ninguna forma mágica para evitar ese requisito en Azure. Lo siento. La verdad es que los SSD superan en gran medida los discos giratorios normales. Hay límites físicos en cuanto a lo rápido que los discos giratorios pueden... bueno, girar. Los ingenieros de Microsoft aún no han sido capaces de cambiar las leyes de la física. Todavía hay casos de uso de discos giratorios normales, como el almacenamiento de archivos de bajo costo, y al igual que en las matrices de almacenamiento normales, las últimas tecnologías pueden proporcionar un buen rendimiento con un grupo de discos giratorios.

La primera y principal elección que debe tomar para una VM Azure es qué tipo de almacenamiento utilizar:

- *Discos SSD premium*: utilice discos SSD de alto rendimiento para obtener un rendimiento óptimo, mayores IOPS y baja latencia; tipo de almacenamiento recomendado para la mayoría de las cargas de trabajo de producción.
- *Discos SSD estándar*: utilice SSD estándar y ofrezca un rendimiento coherente en comparación con las unidades de disco duro (HDD). Estos discos son excelentes para las cargas de trabajo de desarrollo y prueba o para usos de producción de presupuesto reducido o de baja demanda.
- *Discos HDD estándar*: utilice discos giratorios normales para accesos menos frecuentes a los datos, como archivos y copias de seguridad.

El tamaño de VM que elija ayuda a determinar qué tipo de almacenamiento puede seleccionar. En el capítulo 2, cuando creó una VM, escogió un tamaño que le permitía crear rápidamente una VM. El valor predeterminado era probablemente algo como una VM de la serie D2S_v3, que le daba acceso a discos SSD de primera calidad. ¿Cómo

puede saber qué VM pueden acceder a discos SSD premium? Busque una s en el tamaño de VM, para SSD. Hay un par de excepciones a la regla, pero es un buen patrón que seguir. Los siguientes ejemplos le ayudarán a identificar qué máquinas virtuales pueden acceder a discos premium y qué máquinas virtuales pueden acceder a discos SSD o HDD estándar:

- Las máquinas virtuales de las series D2S_v3, Fs, GS y Ls pueden acceder a discos SSD premium.
- Las VM series D, A, F y M solo pueden acceder a discos SSD o HDD estándar.

Si selecciona un tamaño de VM que puede utilizar discos SSD premium, no hay ninguna obligación de hacerlo. Puede crear y utilizar discos SSD o HDD estándar. Al elegir discos SSD premium, asegura la aplicación a futuro y se da la opción de usar SSD de alto rendimiento cuando los necesite sin necesidad de cambiar el tamaño de sus VM y tener tiempos de inactividad en el proceso. Todos los tamaños de VM pueden usar discos SSD estándar.

Disco de SO efímero

Existe un tipo especial de disco del SO llamado *disco efímero*. Sigue siendo un disco administrado, pero es local al host de Azure subyacente. Este hecho hace que un disco efímero sea realmente rápido, con baja latencia.

Como los datos no se escriben en una matriz de almacenamiento remota, es posible que los datos no persistan durante los reinicios de la máquina virtual si se traslada a un host subyacente diferente. Los discos efímeros son ideales para las cargas de trabajo sin estado que pueden manejar el arranque con una imagen limpia cada vez y no necesitan almacenar datos localmente para acceder a ellos a través de los reinicios.

Solo algunos tamaños de máquinas virtuales son compatibles con los discos efímeros, pero su uso no supone ningún costo adicional y están disponibles en todas las regiones. Se pierde algo de funcionalidad para cosas como Azure Site Recovery y Azure Disk Encryption (capítulos 13 y 14, respectivamente), pero si se quiere un almacenamiento de alta velocidad y baja latencia, hay que echar un vistazo a los discos efímeros.

Pruébelo ahora

¿Cómo puede saber qué tamaños de VM están disponibles para usted? En Azure Portal, abra Cloud Shell. Escriba el siguiente comando (puede usar su propia región):

```
az vm list-sizes --location eastus --output table
```

Recuerde que cualquier tamaño con una s le da acceso a discos SSD premium.

4.1.2 Discos temporales y discos de datos

Ahora que sabe qué nivel de rendimiento necesita para sus aplicaciones, vamos a analizar otro par de piezas del rompecabezas. Los discos se conectan de dos maneras:

- *Discos temporales*: cada VM cuenta de forma automática con almacenamiento SSD local conectado desde el host subyacente, que ofrece una pequeña cantidad de almacenamiento de alto rendimiento. Tenga mucho cuidado con cómo utiliza este disco temporal. Como su nombre lo indica, es posible que este disco no persista con la VM. Si la VM se mueve a un nuevo host en un evento de mantenimiento, se adjuntará un nuevo disco temporal y se perderán todos los datos que hayas almacenado allí. El disco temporal está diseñado para que sea un espacio instantáneo o caché de aplicaciones.

- *Discos de datos*: los discos que conecte específicamente a la VM actúan como es de esperar en términos de particiones, sistemas de archivos y puntos de montaje persistentes. Los discos de datos se vuelven a conectar a medida que la VM se desplaza por el centro de datos de Azure, y están donde se almacena la mayor parte de sus aplicaciones y datos. Es posible utilizar espacios de almacenamiento o RAID de software para agrupar discos de datos en la VM para un rendimiento aún mayor.

Existe un tipo específico de disco de datos que puede conectar a una máquina virtual si necesita el máximo rendimiento y una baja latencia: los discos ultra. Estos discos están un paso por encima de los discos SSD premium y están disponibles solo para discos de datos. Los discos Ultra están diseñados para grandes bases de datos y cargas de trabajo intensivas como SAP HANA. ¿De qué nivel de rapidez estamos hablando? En el momento de escribir este artículo, los discos ultra pueden tener un tamaño de hasta 64 TiB y proporcionar hasta 160 000 IOPS por disco con un rendimiento máximo de 2000 MBps.

4.1.3 *Opciones de almacenamiento en caché de disco*

También es importante considerar el disco del sistema operativo que viene con la VM. Cuando se crea una VM, siempre se obtiene al menos un disco: el disco donde se instala el propio SO. Resulta tentador utilizar ese disco para instalar sus aplicaciones o escribir en él archivos de registro. A menos que ejecute una pequeña implementación de prueba de concepto, no ejecute las aplicaciones en el disco del SO, ya que es muy probable que no obtenga el rendimiento que desea.

Los discos en Azure pueden tener configurada una directiva para el almacenamiento en caché. De forma predeterminada, el disco del sistema operativo ha aplicado *lectura/escritura* para almacenamiento en caché. Normalmente, este tipo de almacenamiento en caché no es el ideal para cargas de trabajo de aplicaciones que escriben, por ejemplo, archivos de registro o bases de datos. Por el contrario, los discos de datos tienen una directiva de caché predeterminada de *ninguno*. Esta es una buena directiva para cargas de trabajo que realizan muchas escrituras. También puede aplicar una directiva de caché *solo lectura*, que se adapte mejor a las cargas de trabajo de aplicaciones que principalmente leen los datos de los discos.

En general, siempre conecte y utilice discos de datos para instalar y ejecutar sus aplicaciones. Incluso la directiva de caché predeterminada "ninguno" ofrece probablemente un mejor rendimiento que la directiva de caché de lectura/escritura del disco del SO.

4.2 *Agregar discos a una VM*

En esta sección, verá cómo agregar discos a una máquina virtual a medida que la crea. En el capítulo 2, creó una VM en Azure Portal. Esta vez, usemos la CLI de Azure para crear una VM. La CLI de Azure proporciona una forma rápida de crear una VM y conectar un disco de datos al mismo tiempo.

Pruébelo ahora

Complete los siguientes pasos para crear una VM y ver los discos de datos en acción:

- 1 En Azure Cloud Shell, cree un grupo de recursos con `az group create` y proporcione un nombre para el grupo de recursos junto con una ubicación:

```
az group create --name azuremolchapter4 --location eastus
```

- 2 Cree una VM con el comando `az vm create`. El parámetro final, `--data-disk-sizes-gb`, le permite crear un disco de datos junto con la VM. En el laboratorio de final del capítulo, puede conectarse a esta VM e inicializar los discos.

- Para este ejercicio, puede crear una VM Linux o Windows. Si se siente cómodo con Linux o quiere aprender a inicializar y preparar un disco para Linux, use el siguiente comando para crear una VM de Ubuntu LTS:

```
az vm create \
    --resource-group azuremolchapter4 \
    --name storagevm \
    --image UbuntuLTS \
    --size Standard_B1ms \
    --admin-username azuremol \
    --generate-ssh-keys \
    --data-disk-sizes-gb 64
```

- Si se siente más cómodo con Windows, utilice el siguiente comando para crear una VM Windows Server 2019. Luego, puede utilizar RDP para conectarse a la VM para configurar los discos más adelante:

```
az vm create \
    --resource-group azuremolchapter4 \
    --name storagevm \
    --image Win2019Datacenter \
    --size Standard_B1ms \
    --admin-username azuremol \
    --admin-password P@ssw0rd! \
    --data-disk-sizes-gb 64
```

- Crear una VM demora unos minutos. La VM ya tiene un disco de datos conectado y está listo para utilizarse.

¿Qué ocurre si desea agregar otro disco de datos después de unas semanas o meses? Utilice nuevamente la CLI de Azure para ver cómo agregar rápidamente un disco. El proceso es el mismo para una VM Linux o Windows. Lo único que debe hacer es decirle a Azure que cree un nuevo disco y lo conecte a su VM.

Pruébelo ahora

Agregue un disco de datos adicional a la VM como se muestra a continuación.

Cree un nuevo disco con el comando `az vm disk attach`. Proporcione un nombre y un tamaño para el disco. ¿Recuerda lo que hablamos de los discos estándar y premium? En el ejemplo siguiente, creará un disco SSD premium:

```
az vm disk attach \
    --resource-group azuremolchapter4 \
    --vm-name storagevm \
    --name datadisk \
    --size-gb 64 \
    --sku Premium_LRS \
    --new
```

¿Reconoce la última parte de ese tipo de almacenamiento? *LRS* significa *almacenamiento con redundancia local*. Analizaremos las opciones de redundancia en la sección 4.3.3.

En dos comandos, creó una VM con la CLI de Azure que incluía un disco de datos y simuló cómo conectar un disco de datos adicional posteriormente. No obstante, el solo hecho de haber conectado estos discos no significa que pueda escribir datos inmediatamente. Al igual que con cualquier disco, ya sea un disco físico conectado a un

servidor local o un disco virtual conectado a una VM, es necesario inicializar el disco, crear una partición y un sistema de archivos. Puede hacerlo en el ejercicio opcional del laboratorio de fin de capítulo.

4.3 Azure Storage

El almacenamiento puede no ser un tema obvio para examinar cuando se trata de compilar y ejecutar aplicaciones, pero es un servicio amplio que cubre mucho más de lo que podría esperar. El servicio de Azure Storage ofrece muchas opciones más disponibles que solo un lugar para almacenar archivos o discos virtuales para sus VM.

Revise lo que puede necesitar su negocio de pizzas ficticio para compilar una aplicación que procesa los pedidos de clientes para llevar o entregar. La aplicación necesita un almacén de datos que contenga las pizzas disponibles, la lista de ingredientes, y los precios. A medida que se reciben y procesan los pedidos, la aplicación necesita una manera de enviar mensajes entre los componentes de la aplicación. Luego, la interfaz del sitio web necesita imágenes llamativas para mostrarlas a los clientes cómo se ven las pizzas. Como puede ver en la figura 4.1, Azure Storage tiene una gran variedad de funciones de almacenamiento y puede cubrir estas tres necesidades.

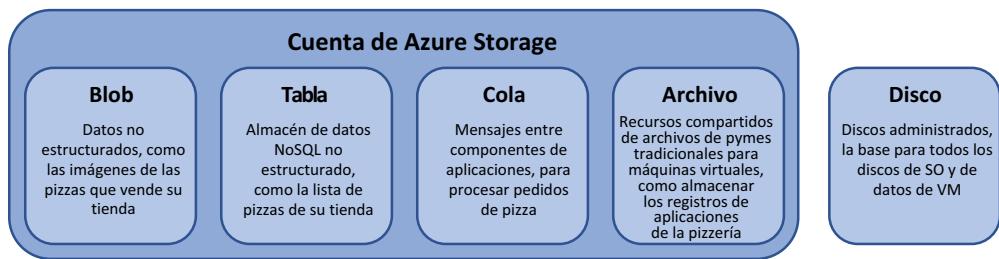


Figura 4.1 Una cuenta de Azure Storage le permite crear y utilizar una amplia variedad de funciones de almacenamiento, mucho más que solo un lugar para almacenar archivos.

- *Almacenamiento de blobs:* para datos no estructurados, como archivos multimedia y documentos. Las aplicaciones pueden almacenar datos en almacenamiento de blobs, como imágenes, para luego representarlas. Usted podría almacenar imágenes de sus pizzas en almacenamiento de blobs.
- *Almacenamiento de tablas:* para datos no estructurados en un almacén de datos NoSQL. Al igual que con cualquier debate sobre los almacenes de datos SQL versus NoSQL, planifique su aplicación y calcule los requisitos de rendimiento para procesar grandes cantidades de datos. Podría almacenar la lista de pizzas de su menú en almacenamiento de tablas. La sección 4.3.1 explora NoSQL con mayor detalle.
- *Almacenamiento de cola:* para que las aplicaciones en la nube se comuniquen entre varios niveles y componentes de manera confiable y coherente. Puede crear, leer y eliminar mensajes que pasen entre los componentes de la aplicación. Puede utilizar el almacenamiento de cola para pasar mensajes entre el front-end web cuando un cliente hace un pedido y el back-end para procesar y hornear las pizzas.

- *Almacenamiento de archivos*: para un recurso compartido de archivos de bloque de mensajes del servidor (SMB) a la vieja usanza, accesible tanto por plataformas Windows como Linux/macOS; a menudo se utiliza para centralizar la recopilación de registros de las máquinas virtuales.

Es fácil utilizar Azure Storage para las VM. Usted crea y utiliza Azure Managed Disks, un tipo de disco duro virtual (VHD) que abstrae muchas consideraciones de diseño relativas al rendimiento y distribuye los discos virtuales a través de la plataforma. Crea una VM, conecta los discos de datos administrados y deja que la plataforma Azure calcule la redundancia y la disponibilidad.

4.3.1 Almacenamiento de tablas

Analicemos un par de tipos de almacenamiento de datos. Primero está el *almacenamiento de tabla*. Probablemente, la mayoría de las personas están familiarizadas con una base de datos SQL tradicional como Microsoft SQL Server, MySQL o PostgreSQL. Se trata de *bases de datos relacionales*, constituidas por una o más tablas que contienen una o más filas de datos. Las bases de datos relacionales son comunes en la compilación de aplicaciones y se pueden diseñar, visualizar y consultar de forma estructurada: la *SQL* (en inglés, Structured Query Language; en español, lenguaje de consulta estructurada).

Las bases de datos NoSQL son un poco diferentes. No siguen el mismo enfoque estructurado y los datos no se almacenan en tablas donde cada fila contiene los mismos campos. Existen diferentes implementaciones de bases de datos NoSQL: algunos ejemplos incluyen MongoDB y CouchDB. Las ventajas conocidas de las bases de datos NoSQL son que escalan horizontalmente (lo que significa que puede agregar más servidores en lugar de agregar más memoria o CPU), pueden gestionar cantidades de información más grandes y son más eficientes en el procesamiento de esos grandes conjuntos de datos.

La forma en que se almacenan los datos en una base de datos NoSQL se puede definir en algunas categorías:

- *Claves/valores*, como Redis
- *Columnas*, como Cassandra
- *Documentos*, como MongoDB

Cada enfoque tiene pros y contras desde un punto de vista de rendimiento, flexibilidad o complejidad. Una tabla de almacenamiento Azure utiliza un almacén de claves/valores y es una buena introducción a las bases de datos NoSQL cuando está acostumbrado a una base de datos SQL como Microsoft SQL o MySQL.

Puede descargar e instalar Azure Storage Explorer en <https://azure.microsoft.com/features/storage-explorer> si desea visualizar los datos. No necesita hacerlo en este momento. Storage Explorer es una gran herramienta para aprender cómo se ven en acción las tablas y las colas. En este capítulo, no queremos profundizar demasiado en las bases de datos NoSQL; en el capítulo 10 se exploran en profundidad algunas bases de datos NoSQL con Azure Cosmos DB. De hecho, en el siguiente ejercicio, deberá utilizar la API de Cosmos DB para conectarse a Azure Storage y crear una tabla. El uso de las tablas Azure es más una introducción a las bases de datos NoSQL que un ejemplo sólido de uso en producción.

Por ahora, vamos a ejecutar una aplicación de ejemplo rápido para ver cómo se pueden agregar y consultar los datos, al igual que lo haría con una aplicación real. Estos ejemplos son básicos pero muestran cómo puede almacenar los tipos de pizzas que vende y cuánto cuesta cada pizza. En lugar de utilizar algo grande como Microsoft SQL Server o MySQL, utilice una base de datos NoSQL con almacenamiento de tabla Azure.

Pruébelo ahora

Complete los siguientes pasos para ver las tablas Azure en acción:

- 1 Abra Azure Portal en un navegador web y, a continuación, abra Cloud Shell.
- 2 En el capítulo 3, obtuvo una copia de los ejemplos de Azure de GitHub. Si no lo hizo, consiga una copia de la siguiente manera:

```
git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
```

- 3 Cámbiese al directorio que contiene los ejemplos de Azure Storage:

```
cd ~/azure-mol-samples-2nd-ed/04
```

- 4 Instale un par de dependencias de Python, si no están ya instaladas. Aquí se instala el paquete azurerm, que maneja la comunicación que le permite crear y administrar recursos de Azure, y dos paquetes azure, que son los SDK de Python subyacentes para Azure Cosmos DB y Storage:

```
pip install --user azurerm azure-cosmosdb-table azure-storage-  
queue==2.1.0
```

¿Qué significa --user cuando instala los paquetes? Si utiliza Azure Cloud Shell, no puede instalar paquetes en el sistema central, ya que no tiene permiso. En lugar de ellos, los paquetes se instalan en el entorno del usuario. Estas instalaciones de paquetes persisten entre sesiones y le permiten utilizar todos los SDK de Azure en estos ejemplos.

- 5 Ejecute la aplicación de ejemplo Python para tablas. Siga las notificaciones para disfrutar de una pizza:

```
python storage_table_demo.py
```

Serpientes en un avión

Python es un lenguaje de programación ampliamente utilizado que a menudo se utiliza en las clases de "Introducción a informática". Si trabaja principalmente en operaciones o administración de TI, piense en Python como un potente lenguaje de scripts que funciona a través de los sistemas operativos. Python no solo se usa para scripts, también se puede utilizar para compilar aplicaciones complejas; por ejemplo, la CLI de Azure que ha estado usando está escrita en Python.

Yo utilizo Python para algunos de los ejemplos en este libro, porque debieran funcionar fuera de Cloud Shell sin modificaciones. Las distros de macOS y Linux incluyen Python de forma nativa. Los usuarios de Windows pueden descargar e instalar rápidamente Python y ejecutar estos scripts localmente. Python es ideal para quienes tienen poca o nula experiencia de programación, así como para desarrolladores familiarizados con otros lenguajes. La documentación de Azure para Azure Storage y muchos otros servicios es compatible con una amplia gama de lenguajes, como .NET, Java y Node.js. No está limitado a usar Python cuando compila sus propias aplicaciones que usan tablas.

El libro Quick Python Book, tercera edición, de Naomi Veder (<http://mng.bz/6QZA>), puede ayudarlo a ponerse al día si desea obtener más información. También hay un curso basado en video para *Get Programming with Python in Motion*, de Ana Bell (<http://mng.bz/oPap>).

4.3.2 Almacenamiento de cola

Las tablas Azure son fantásticas cuando comienza a meter las manos en el mundo del desarrollo de aplicaciones en la nube. A medida que comienza a compilar y administrar aplicaciones de forma nativa en la nube, normalmente divide la aplicación en componentes más pequeños, cada uno de los cuales puede escalar de forma independiente y procesar datos por cuenta propia. Para permitir que estos distintos componentes se comuniquen y pasen datos de un lado a otro, se suele requerir alguna forma de cola de mensajes. Escriba Azure Queues.

El servicio de Azure Queues le permite crear, leer y luego eliminar mensajes que conllevan pequeños trozos de datos. Estos mensajes son creados y recuperados por diferentes componentes de la aplicación a medida que los datos pasan de un lado a otro. Azure Queues no eliminará un mensaje hasta que una aplicación haya terminado de procesar los datos del mensaje.

Pruébelo ahora

Para ver Azure Queues en acción, ejecute el siguiente script de Python desde el mismo directorio azure-samples/4. Siga las notificaciones para ver los mensajes escritos, leídos y eliminados de la cola:

```
python storage_queue_demo.py
```

Continúe con la aplicación de ejemplo que gestiona los pedidos de pizza. Es posible que tenga un componente de aplicación front-end con el que los clientes interactúen para pedir su pizza y, a continuación, una cola de mensajes que transmita mensajes a un componente de la aplicación de back-end que procesa esos pedidos. A medida que se reciben los pedidos, los mensajes en la cola se pueden visualizar, como se muestra en la figura 4.2.

ID	Message Text	Insertion Time (UTC)	Expiration Time (UTC)	Dequeue Count	Size
ca57a12c-21b8-4640-9e07-4fc3a81c8dd5	Veggie pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	21 B
d68f90a9-1d5a-4a0e-aef79-f285efa2aca2	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B
7f3c6f4a-9d47-488f-9344-1cb5bbec0fa4	Hawlian pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	22 B
63f07e06-ab0d-48c0-81c9-019d4255f335	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B
66b48f73-d136-4d82-9b41-f32f93f3d725	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B

Figura 4.2 Los mensajes se reciben desde el componente de aplicación front-end que detalla la pizza que cada cliente pidió en la propiedad `Message Text`.

A medida que el componente de la aplicación de back-end procesa cada pedido de pizza, los mensajes se van eliminando de la cola. En la figura 4.3 se muestra cómo se ve la cola cuando tiene una pizza vegetariana en el horno y ese primer mensaje se elimina.

ID	Message Text	Insertion Time (UTC)	Expiration Time (UTC)	Dequeue Count	Size
d68f90a9-1d5a-4a0e-aef79-f285efa2aca2	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B
7f3c6f4a-9d47-488f-9344-1cb5bbec0fa4	Hawlian pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	22 B
63f07e06-ab0d-48c0-81c9-019d4255f335	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B
66b48f73-d136-4d82-9b41-f32f93f3d725	Pepperoni pizza ordered.	Fri, 23 Aug 2019 03:39:39 GMT	Fri, 30 Aug 2019 03:39:39 GMT	0	24 B

Figura 4.3 A medida que se procesa cada mensaje, se va eliminando de la cola. El primer mensaje que se muestra en la figura 4.2 se eliminó después de que fue procesado por el componente de la aplicación de back-end.

4.3.3 Disponibilidad y redundancia de almacenamiento

Los centros de datos de Azure están diseñados para tolerar fallas de conexiones de Internet redundantes, generadores de energía, múltiples rutas de red, matrices de almacenamiento, etc. Sin embargo, aún necesita hacer su parte cuando diseña y ejecuta aplicaciones. Con Azure Storage, usted elige el nivel de redundancia de almacenamiento que necesita. Este nivel varía para cada aplicación y según la importancia de los datos. Estas son las opciones de redundancia de almacenamiento disponibles:

- *Almacenamiento con redundancia local (LRS)*: sus datos se replican tres veces dentro del único centro de datos donde se creó su cuenta de almacenamiento. Esta opción proporciona redundancia en el caso de una falla de hardware única, pero si se cae todo el centro de datos (raro, pero posible), su información también se cae.
- *Almacenamiento con redundancia de zona (ZRS)*: el siguiente nivel de LRS replica sus datos tres veces en dos o tres centros de datos en una región (cuando existen varios centros de datos en una región) o en todas las regiones. ZRS también está disponible en distintas zonas de disponibilidad, las que exploraremos en el capítulo 7.
- *Almacenamiento con redundancia geográfica (GRS)*: con GRS, sus datos se replican tres veces en la región principal en la que se crea el almacenamiento y luego se replica tres veces en una región sincronizada. La región sincronizada suele estar a cientos o más kilómetros de distancia. Por ejemplo, la Oeste de EE. UU. está

sincronizada con la Este de EE. UU., Europa del Norte está sincronizado con Europa Occidental, y el Sudeste Asiático está sincronizado con Asia Oriental. GRS ofrece una gran opción de redundancia para aplicaciones de producción.

- *Almacenamiento con redundancia geográfica con acceso de lectura (RA-GRS)*: esta es la opción de redundancia de datos premium. Sus datos se replican en regiones sincronizadas como GRS, pero también tiene acceso de lectura a los datos en esa zona secundaria.

4.4 Laboratorio: Explorar Azure Storage

Aquí tiene la oportunidad de probar sus habilidades. Escoja una de las siguientes tareas para completar el ejercicio de laboratorio.

4.4.1 Centrado en la VM

Si desea iniciar sesión en una VM y ver que el proceso para inicializar un disco y crear un sistema de archivos es el mismo que en cualquier otra VM con la que ha trabajado, pruebe uno de estos ejercicios:

- 1 Inicie sesión en la VM que creó en la sección 4.2. Dependiendo de lo que elija, se conectará mediante SSH o RDP.
- 2 Inicialice el disco y cree una partición.
 - En Linux, el flujo es `fdisk`, `mkfs` y luego `mount`.
 - En Windows, utilice cualquier secuencia con la que se sienta cómodo, probablemente Administración del disco > Inicializar > Crear volumen > Formato.

4.4.2 Centrado en el desarrollador

Si su perfil es más desarrollador y no quiere saber cómo inicializar discos de datos en una VM, regrese a Cloud Shell y explore las dos demostraciones de Python que utilizan tablas y colas. Aunque sea nuevo en Python, debiera ser capaz de seguir lo que está pasando:

- Piense en algunas situaciones en las que podría implementar tablas o colas en sus propias aplicaciones. ¿Qué se necesitaría para compilar aplicaciones nativas en la nube con componentes de aplicación individuales que puedan utilizar colas, por ejemplo?
- Modifique uno de los ejemplos de su interés para crear un elemento adicional para el menú de pizza (si es una tabla) o cree un nuevo mensaje de pedido de pizza (si es una cola).

Conceptos básicos de las redes de Azure

En el capítulo 4, exploró el servicio de Azure Storage. Uno de los otros servicios básicos para las aplicaciones en la nube es Redes de Azure. Azure tiene un montón de potentes funciones de red para asegurar y enrutar su tráfico a una escala verdaderamente global. Estas funciones están diseñadas para ayudarle a centrarse en cómo compilar y mantener sus aplicaciones, de modo que no tenga que preocuparse por detalles como direcciones IP y redirección de tablas. Si compila y ejecuta una tienda en línea para gestionar pedidos de pizza, debe transmitir los datos del cliente y procesar las transacciones de pago de forma segura.

En este capítulo se examinan las redes y subredes virtuales de Azure, y se analiza cómo crear interfaces de red virtuales. Para asegurar y controlar el flujo de tráfico, cree grupos y reglas de seguridad de red. Si la red es nueva para usted, o si hace tiempo que no tiene que trabajar con direcciones IP y tarjetas de red, puede que este capítulo le tome un poco más de tiempo. Tiene muchos ejercicios Pruébelo ahora. Sin embargo, vale la pena que se tome el tiempo para entender este capítulo, ya que el trabajo en red es clave para muchos de los servicios de Azure.

5.1 Componentes de las redes virtuales

Piense en cuántos cables hay detrás del escritorio de su equipo o en su centro de entretenimiento. Ahora, piense en todos los cables necesarios para conectar los equipos en un piso dado de un edificio de oficinas. ¿Y qué pasa con el edificio de oficinas en su totalidad? ¿Ha estado alguna vez en un centro de datos o ha visto fotos de uno? Intente imaginar lo grande que son los centros de datos de Azure. Ahora intente imaginar docenas de centros de datos Azure en todo el mundo. Las matemáticas no son mi punto fuerte, así que ni siquiera puedo calcular cuántos kilómetros y kilómetros de cables de red se utilizan para llevar todo el tráfico en Azure.

La conectividad de red es una parte crucial de la vida moderna. En Azure, la red es fundamental para la forma en que se comunica todo. Para todos los miles de dispositivos de red físicos y kilómetros de cables de red que conectan todo en un centro de datos de Azure, se trabaja con recursos de red *virtuales*. ¿Cómo? Redes definidas por software. Cuando crea una VM o una aplicación web, no es necesario que un técnico corra por el centro de datos de Azure para conectar cables físicamente ni asignar direcciones IP (¡aunque sería divertido de ver!). En lugar de ello, la plataforma Azure maneja lógicamente todos los recursos de red que definen todo su entorno de red. En la figura 5.1 se muestran los componentes de red virtuales que compilará a medida que trabaje en este capítulo.

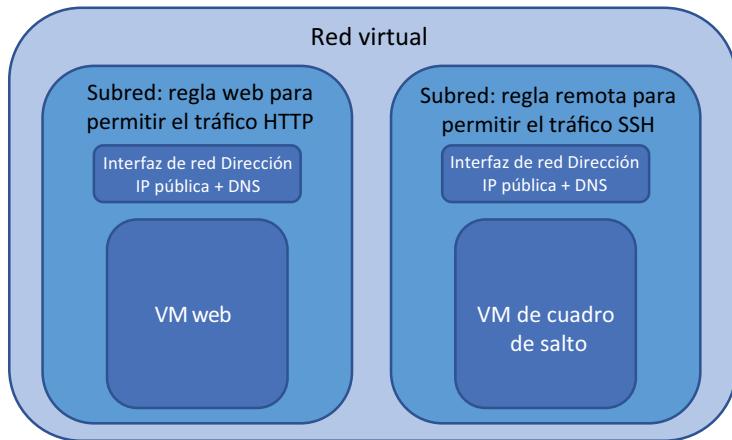


Figura 5.1 Conexiones de red definidas por software en Azure

Algunos de los componentes de red se abstraen si utiliza recursos PaaS. Los componentes principales que utiliza para las VM son los siguientes:

- Redes y subredes virtuales (incluidas las agrupaciones de direcciones IP).
- Tarjetas de interfaz de red virtuales.
- Una o más direcciones IP públicas.
- Nombre DNS interno y nombres DNS públicos opcionales para resolución de nombres externos.
- Reglas y grupos de seguridad de red, que aseguran y controlan el flujo de tráfico de red como lo hace un firewall normal.

5.1.1 Redes y subredes virtuales

Cuando creó una VM en el capítulo 2, no tuvo que ajustar ningún parámetro de red. La plataforma Azure puede crear estos recursos automáticamente con nombres y ámbitos de dirección IP predeterminados. En esta sección, creará los recursos de red con antelación y veamos cómo funcionan para una VM.

Pruébelo ahora

La creación de redes suele ser más fácil de visualizar cuando se ve en acción. Utilizará Azure Portal para empezar, lo que acaba suponiendo bastantes pasos separados, pero verá la potencia de la CLI de Azure más adelante en el capítulo.

No se preocupe demasiado sobre cómo utilizar sus propios espacios de direcciones o nombres de DNS personalizados en este momento. Complete los siguientes pasos para crear la red y subred virtuales:

- 1 Abra Azure Portal y seleccione Crear un recurso en la esquina superior izquierda del panel.
- 2 Seleccione Redes en la lista de servicios de Marketplace y, a continuación, seleccione Red virtual.
- 3 Escriba un nombre para la red virtual, como vnetmol.
- 4 Si quiere jugar un poco más, cambie la dirección a 10.0.0.0/16.

Intervalos de direcciones IP

Las redes virtuales abarcan cierto intervalo de IP: un espacio de direcciones. Si alguna vez ha visto una dirección IP, puede haberse fijado en la máscara de subred que, a menudo, es algo como 255.255.255.0. Esta máscara de subred suele utilizarse en un formato corto que especifica el tamaño del intervalo, como por ejemplo /24.

Azure Portal toma un espacio de direcciones /24 predeterminado. Aquí aumentaremos el número de intervalos IP adicionales sin demasiado conocimiento de redes, así que aumente el espacio de dirección a /16. Este tipo de dirección IP no se proporciona directamente a una VM; en el siguiente paso, creará una subred que cubre una sección más pequeña de este espacio de dirección.

Si los espacios de direcciones de red son totalmente ajenos para usted, no se preocupe; en su mayoría, no lidiará con ellos a diario. La gobernanza adecuada de Azure puede funcionar de la misma manera que lo hace en su mundo de TI local: un grupo de personas puede administrar las redes virtuales de Azure y usted implementa sus aplicaciones en un espacio creado previamente.

- 5 Cree un grupo de recursos, como azuremolchapter5 y, luego, seleccione una región de Azure cerca de usted.
- 6 Proporcione un nombre de subred, como websubnet, y escriba el intervalo de direcciones de subred 10.0.1.0/24. Este intervalo de direcciones forma parte de la red virtual más amplia especificada anteriormente. Más tarde, agregará otra subred.
- 7 Examine algunas de las otras opciones, como la protección contra la denegación de servicio distribuida (DDoS), los puntos de conexión de servicio y Azure Firewall. Deje los valores predeterminados por ahora, pero espero que este ejemplo le dé algunas pistas sobre lo que es posible más allá de una red virtual básica.
- 8 Cuando esté listo, cree la red y subred virtuales.

5.1.2 Tarjetas de interfaz de red virtuales.

Ahora que creó una red y subred virtuales, necesita conectarse una VM. Al igual que lo hace con un equipo de escritorio normal, un equipo portátil o una tableta, se utiliza una tarjeta de interfaz de red (NIC) para conectarse a la red virtual. Y no, ¡no hay Wi-Fi gratuito! Pero hay tamaños de VM en Azure que actualmente proporcionan hasta ocho NIC con velocidades de hasta 32 GBps. Aunque se me dieran bien las matemáticas, podría decir que estas cifras suman un gran ancho de banda.

Se puede estar preguntando por qué es necesario crear cada uno de estos recursos con anticipación. Puede hacer todo esto en el momento de crear una VM. Es cierto... pero dé un paso atrás y piense en los recursos de red como recursos de larga duración.

Los recursos de red existen de forma separada de los recursos de la VM, y pueden persistir más allá del ciclo de duración de una VM determinada. Este concepto le permite crear los recursos de red fijos y crear, eliminar y crear nuevamente una VM que conserve los mismos recursos de red, como direcciones IP y nombres DNS. Piense en una VM de laboratorio o en un entorno de desarrollo y prueba. Puede reproducir exactamente el mismo entorno de forma rápida, ya que solo cambia la VM.

Pruébelo ahora

Para crear una NIC, complete los pasos siguientes:

- 1 En Azure Portal, seleccione Crear un recurso en la esquina superior izquierda del panel.
- 2 Busque y seleccione Interfaz de red y, a continuación, seleccione Crear.
- 3 Proporcione un nombre para la interfaz de red, como webvnic; luego, seleccione la red y subred virtuales que creó en el ejercicio anterior.
- 4 Antes he hablado de los recursos de larga duración; ahora puede ver cómo funcionan. Cree una asignación de dirección IP estática que utilice la dirección 10.0.1.4.

CONSEJO ¿Por qué .4? ¿Qué pasa con las primeras tres direcciones en el espacio de direcciones? Azure reserva las primeras tres direcciones IP en cada intervalo para su propia administración y enrutamiento. La primera dirección que se puede utilizar en cada intervalo es .4.

- 5 No cree un grupo de seguridad de red por ahora; volverá a esto en unos minutos. Si es experto en IPv6, puede activar el cuadro de Dirección IP privada (IPv6) y escribir un nombre; de lo contrario, continúe con IPv4.
- 6 Seleccione el grupo de recursos existente del ejercicio anterior y, a continuación, elija crear la NIC en la misma región que la red virtual.
- 7 Cuando esté listo, cree la NIC.

Separación de funciones en Azure

No tiene que crear otros procesos dentro del mismo grupo de recursos que su red virtual. Vuelva a pensar en el concepto de gobernanza de Azure que se analizó anteriormente. Puede tener un grupo de ingenieros de red que administren todos los recursos de red virtual en Azure. Cuando se crean recursos para las aplicaciones, como las VM, los crea y administra en sus propios grupos de recursos.

En capítulos posteriores se analizan algunas de las funciones de seguridad y directiva en Azure que le permiten definir quién puede acceder y editar ciertos recursos. La idea es que si no sabe, o no quiere saber mucho sobre los recursos de la red, puede conectarse a lo que se le asignó y listo. Lo mismo se aplica a otros ingenieros o desarrolladores: es posible que puedan visualizar los recursos de su aplicación, pero no editarlos ni eliminarlos.

Este tipo de modelo de gobernanza en Azure es cómodo, pero tenga cuidado de evitar caer en la trampa de trabajar en silos. En las grandes empresas, puede ser inevitable que esté limitado por las líneas de departamentos. Sin embargo, una de las grandes ventajas de los proveedores de informática en la nube como Azure es acelerar el tiempo de implementación de las aplicaciones, ya que no hay que esperar el cableado y la configuración de los recursos de una red física. Planifique la creación y configuración de los recursos de red de Azure, y luego debería ser capaz de crear y administrar sus recursos de aplicación sin problemas.

5.1.3 Dirección IP pública y resolución DNS

Nadie puede acceder a sus recursos todavía, porque no hay direcciones IP públicas o nombres DNS asociados a ellos. Insisto, siga el principio de los recursos de larga duración para crear una dirección IP pública y un nombre DNS público y, a continuación, asignémoslos a su interfaz de red.

Pruébelo ahora

Complete los siguientes pasos para crear una dirección IP pública y una entrada DNS para su interfaz de red:

- 1 En Azure Portal, seleccione Crear un recurso en la esquina superior izquierda del panel.
- 2 Busque y seleccione Dirección IP pública y, a continuación, seleccione Crear.
- 3 Cree una SKU básica y una dirección IPv4. Las SKU estándar y las direcciones IPv6 se utilizan con los equilibradores de carga (capítulo 8). No se preocupe demasiado por la diferencia en este momento.
- 4 Escriba un nombre, como webpublicip, que use una asignación dinámica.

Tipos de asignación de direcciones IP

Una asignación dinámica asigna una dirección IP pública cuando se inicia la VM. Cuando se detiene la VM, se desasigna la dirección IP pública. Hay un par de puntos importantes aquí:

- No tendrá una dirección IP pública hasta que la asigne a una VM y la inicie.
- La dirección IP pública puede cambiar si detiene, desasigna e inicia la VM.

Una asignación estática significa que tiene una dirección IP pública asignada sin una VM asociada, y esa dirección no cambiará. Esta asignación es útil cuando utiliza un certificado SSL asignado a una dirección IP, o un nombre DNS personalizado y un registro que dirige a la dirección IP.

Ahora mismo, está usando una sola VM. Para usos de producción, idealmente ejecutará su aplicación en varias VM con un equilibrador de carga delante de ellos. En esa situación, la dirección IP pública se asigna al equilibrador de carga y normalmente crea una asignación estática en ese punto.

- 5 Escriba un nombre DNS único. Este nombre forma el nombre de dominio completo (FQDN) para su recurso, que se basa en la región de Azure en la que se crea. Si, por ejemplo, crea un nombre de DNS llamado `azuremol` en la región Este de EE. UU., el FQDN pasa a ser `azuremol.eastus.cloudapp.azure.com`.

Entradas DNS

¿Qué pasa con un nombre DNS personalizado? El FQDN predeterminado no es exactamente fácil de usar. Utilice una dirección IP pública estática y, a continuación, cree un registro CNAME en su zona DNS registrada. Usted mantiene el control del registro DNS y puede crear tantas entradas como desee para sus aplicaciones.

Como ejemplo, en la zona DNS `manning.com`, puede crear un CNAME para `azuremol` que dirija a una dirección IP pública estática en Azure. Un usuario tendría que acceder a `azuremol.manning.com` para llegar a su aplicación. Esta dirección es mucho más fácil de usar que `webmol.eastus.cloudapp.azure.com`.

- 6 Seleccione el grupo de recursos existente del ejercicio anterior y, a continuación, elija crear la dirección IP pública en la misma región que la red virtual.
- 7 Cuando esté listo, cree la dirección IP pública.
- 8 Asocie la dirección IP pública y la etiqueta de nombre DNS con la interfaz de red que creó en la sección 5.1.2. Busque y seleccione Grupos de recursos en la barra de navegación del lado izquierdo de Azure Portal. Luego, elija el grupo de recursos en el que creó los recursos de red, como `azuremolchapter5`.
- 9 Seleccione la dirección IP pública de la lista de recursos y, a continuación, elija Asociar.
- 10 Elija asociar con una interfaz de red (pero fíjese con qué otra cosa puedes asociar la dirección IP pública); luego elija la interfaz de red que creó, como `webvnic`.

Después de unos segundos, la dirección IP pública se actualiza para mostrar que la dirección IP ya está asociada a su interfaz de red. Si seleccionó Dinámico como tipo de asignación, la dirección IP seguirá estando en blanco, como se muestra en la figura

5.2. Recuerde que una dirección IP pública se asigna cuando se enciende una VM asociada.

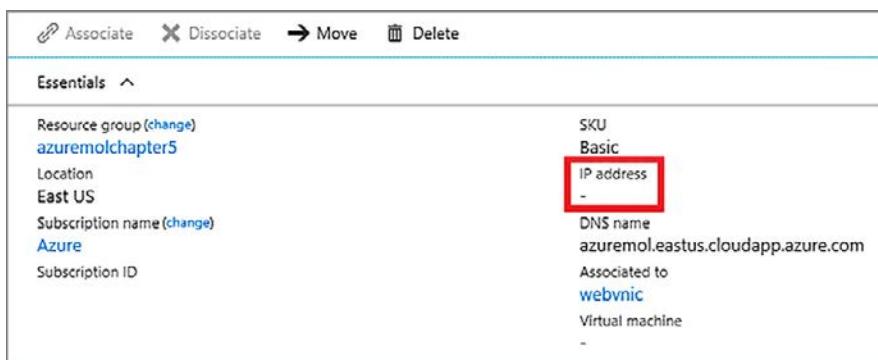


Figura 5.2 La dirección IP pública está asociada a una interfaz de red. Con una asignación dinámica, no se muestra ninguna dirección IP pública hasta que se crea una VM y se enciende.

5.2 Protección y control de tráfico con grupos de seguridad de red

Prueba sorpresa: para controlar y restringir el flujo de tráfico, ¿debe conectar una VM a Internet sin un firewall? Si contestó, "claro, ¿por qué no?" tal vez deba tomarse el resto de su descanso para almorzar para leer un poco acerca de la seguridad en la red informática mundial.

Espero que su respuesta haya sido un rotundo, "*¡no!*". Lamentablemente, su VM tiene mucho potencial como para que sufra un ataque cibernetico automatizado poco después de encenderla. Siempre debe seguir las prácticas recomendadas para mantener actualizados el sistema operativo y el software de la aplicación, pero ni siquiera conviene que el tráfico llegue a la VM si no es necesario. Un equipo de macOS o Windows normal tiene un firewall de software incorporado y cada red local (competente) que he visto tiene un firewall de red entre Internet y la red interna. En Azure, el firewall y las reglas de tráfico son proporcionados por grupos de seguridad de red.

5.2.1 Creación de un grupo de seguridad de red

En Azure, un NSG aplica de manera lógica un conjunto de reglas a los recursos de red. Estas reglas definen qué tráfico puede ingresar y salir de su VM. Usted define los puertos, los protocolos y las direcciones IP que se permiten, y en qué dirección. Estos grupos de reglas se pueden aplicar a una sola interfaz de red o a toda una subred de red. Esta flexibilidad le permite controlar finamente cómo y cuándo se aplican las reglas para satisfacer las necesidades de seguridad de su aplicación.

La figura 5.3 muestra el flujo lógico de un paquete de red entrante cuando pasa a través de un NSG. El mismo proceso se aplicaría a los paquetes salientes. El host Azure no distingue entre el tráfico de Internet y el tráfico de otros lugares dentro de su

entorno Azure, como otra subred o red virtual. Las reglas de NSG entrantes se aplican cualquier paquete de red entrante y las reglas de NSG salientes se aplican a cualquier paquete de red saliente.

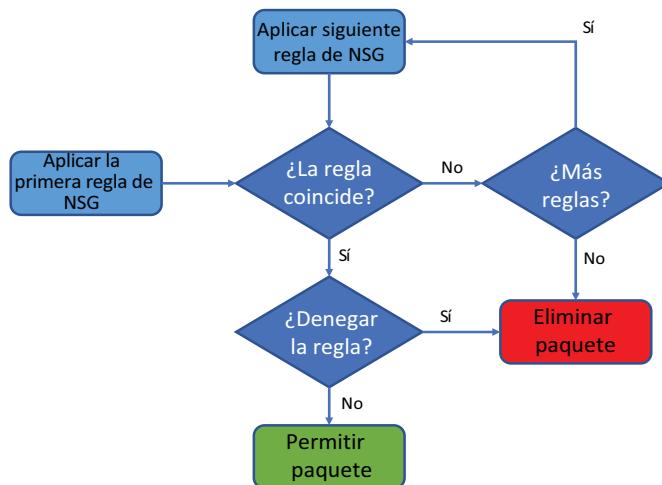


Figura 5.3 Los paquetes entrantes se examinan y cada regla de NSG se aplica en orden de prioridad. Si se produce una coincidencia de regla para Permitir o Denegar, el paquete se reenvía a la VM o se elimina.

Esto es lo que pasa con cada paquete de red:

- 1 Se aplica la primera regla de NSG.
- 2 Si la regla no coincide con el paquete, se carga la siguiente regla hasta que no haya más reglas. A continuación, se aplica la regla predeterminada para eliminar el paquete.
- 3 Si una regla coincide, compruebe si la acción es denegar el paquete. Si es así, el paquete se elimina.
- 4 De lo contrario, si la regla es permitir el paquete, el paquete se pasa a la VM.

A continuación, creará una NGS para que estos conceptos empiecen a tener sentido.

Pruébelo ahora

Complete los siguientes pasos para crear un grupo de seguridad de red:

- 1 En Azure Portal, seleccione Crear un recurso en la esquina superior izquierda del panel.
- 2 Busque y seleccione el Grupo de seguridad de red y, a continuación, seleccione Crear.
- 3 Escriba un nombre, como webnsg, y elija usar el grupo de recursos existente.

¡Listo! La mayor parte de la configuración de un NSG viene cuando se crean las reglas de filtrado. En la sección 5.2.2 se analiza cómo hacer eso y poner su NSG en funcionamiento.

5.2.2 Asociación de un grupo de seguridad de red con una subred

El NSG no hace mucho para proteger sus VM si no tiene ninguna regla. También es necesario asociarlo con una subred, de la misma manera que anteriormente asoció la dirección IP pública con una interfaz de red. Primero asociará su NSG con una subred.

Pruébelo ahora

Complete los siguientes pasos para asociar su subred de red virtual al grupo de seguridad de red:

- 1 Busque y seleccione Grupos de recursos en la barra de navegación del lado izquierdo de Azure Portal. Luego, elija el grupo de recursos en el que creó los recursos de red, como azuremolchapter5.
- 2 Seleccione su NSG, como webnsg.
- 3 A la izquierda, en las opciones de Configuración, se encuentran las Interfaces de red y Subredes. Seleccione Subredes.
- 4 Seleccione el botón Asociar; seleccione la red virtual y la subred de red que creó con anterioridad y, a continuación, seleccione Aceptar para asociar su NSG con la subred.

La flexibilidad de los NSG significa que puede asociar varias subredes, a través de varias redes virtuales, con un solo NSG. La asignación es de uno a muchos, que permite definir reglas básicas de seguridad de red que se aplican a una amplia gama de recursos y aplicaciones.

Ahora puede ver su NSG y qué reglas predeterminadas se aplican.

- 5 En el lado izquierdo de su NSG, seleccione Reglas de seguridad entrantes. No se enumeran reglas de seguridad, al menos ninguna que haya creado.
- 6 Seleccione Reglas predeterminadas para ver lo que la plataforma Azure crea automáticamente, como se muestra en la figura 5.4.

Default rules							
PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION	
65000	AllowVnetInbound	Any	Any	VirtualNetwork	VirtualNetwork	<input checked="" type="checkbox"/> Allow	...
65001	AllowAzureLoadBalancerInbound	Any	Any	AzureLoadBalancer	Any	<input checked="" type="checkbox"/> Allow	...
65500	DenyAllInbound	Any	Any	Any	Any	<input checked="" type="checkbox"/> Deny	...

Figura 5.4 Se crean reglas de seguridad predeterminadas que permiten el tráfico interno de la red virtual o del equilibrador de carga, pero deniegan el resto del tráfico.

Se han creado automáticamente tres reglas predeterminadas, las cuales son importantes de entender:

- *AllowVnetInBound*: permite cualquier tráfico que sea interno a la red virtual. Si tiene varias subredes en su red virtual, el tráfico no se filtra de forma predeterminada y se permite.
- *AllowAzureLoadBalancerInBound*: permite que cualquier tráfico de un equilibrador de carga Azure llegue a su VM. Si coloca un equilibrador de carga entre las VM e Internet, esta regla asegura que el tráfico del equilibrador de carga pueda llegar a sus VM, así como para supervisar el latido.
- *DenyAllInBound*: la regla final que se aplica. Elimina cualquier paquete entrante que llegue hasta este punto. Si antes no hay ninguna regla Permitir previa, esta regla elimina todo el tráfico de forma predeterminada, lo que significa que simplemente debe permitir cualquier tráfico específico que desee; el resto se elimina.

La prioridad de una regla de NSG es importante. Si se aplica una regla Permitir o Denegar, no se aplicarán reglas adicionales. Las reglas se aplican en orden de prioridad numérica ascendente; una regla con una prioridad de 100 se aplica antes de una regla de prioridad 200, por ejemplo.

Al igual que lo conversado anteriormente sobre la gobernanza de los recursos de Azure, puede que ya se hayan creado estas reglas de NSG y se hayan aplicado a una subred determinada. Usted crea sus VM y ejecuta sus aplicaciones, mientras que otra persona administra los NSG.

Es importante entender cómo fluye el tráfico en caso de que algo salga mal. Hay un par de herramientas en Azure que pueden ayudarlo a determinar por qué el tráfico no puede llegar a su aplicación cuando cree que sí debería.

5.2.3 Creación de reglas de filtrado de grupo de seguridad de red

Ahora que tiene su NSG asociado a la subred de la red y hemos revisado las reglas predeterminadas, cree una regla de NSG básica que permita el tráfico HTTP.

Pruébelo ahora

Complete los siguientes pasos para crear sus propias reglas con el grupo de seguridad de red:

- 1 Para crear una regla de NSG desde la ventana anterior de Azure Portal, seleccione Agregar en la sección Reglas de seguridad entrantes.
- 2 Tiene dos opciones para crear reglas: Básica y Avanzada. Para crear rápidamente reglas precompiladas, seleccione Básico en la parte superior de la ventana.
- 3 Elija HTTP en el menú desplegable Servicio. Se proporcionan muchos servicios predeterminados, como SSH, RDP y MySQL. Cuando se selecciona un servicio, se aplica el rango de puertos apropiado, en este caso, el puerto 80. La acción por defecto en las reglas básicas permite el tráfico.

- 4 Se asigna un valor de prioridad a cada regla. Cuanto menor sea el número, mayor será la prioridad. Acepte la prioridad baja predeterminada, como por ejemplo 100.
- 5 Acepte el nombre predeterminado o escriba el que desee; luego, seleccione Aceptar.

5.3 Compilación de una aplicación web de ejemplo con tráfico seguro

Hasta ahora ha creado una red y una subred virtual. Luego, creó una interfaz de red y asoció una dirección IP pública y una etiqueta de nombre DNS. Creó un NSG y lo aplicó a toda la subred, y creó una regla de NSG para permitir el tráfico HTTP. Le falta una cosa: la VM.

5.3.1 Creaciones de conexiones de red de acceso remoto

En producción, no debe abrir el acceso remoto, como SSH o RDP, a las VM que ejecutan sus aplicaciones. Normalmente, tiene una VM de servidor de salto independiente a la que se conecta desde Internet y luego accede a VM adicionales a través de la conexión interna. Hasta ahora, ha creado todos los recursos de red virtual en Azure Portal. Pasemos a la CLI de Azure para ver lo rápido que se pueden crear estos recursos desde la línea de comandos.

Pruébelo ahora

Creó el primer NSG se creó en Azure Portal. Complete los siguientes pasos para crear otro NSG con la CLI de Azure:

- 1 Seleccione el icono de Cloud Shell en la parte superior del panel de Azure Portal. Asegúrese de que se abra el Bash Shell, y no PowerShell.
- 2 Cree un NSG adicional en el grupo de recursos existente. Como en los capítulos anteriores, las barras inversas (\) en los ejemplos de comandos siguientes son para ayudar con los saltos de línea: no tiene que escribirlos si no lo desea. Escriba un nombre, como remotensg:

```
az network nsg create \
--resource-group azuremolchapter5 \
--name remotensg
```

- 3 Cree una regla NSG en el nuevo NSG que *permite* el puerto 22. Escriba el grupo de recursos y NSG que creó en el paso anterior, junto con un nombre, como allowssh:

```
az network nsg rule create \
--resource-group azuremolchapter5 \
--nsg-name remotensg \
--name allowssh \
--protocol tcp \
--priority 100 \
--destination-port-range 22 \
--access allow
```

- 4 Cree una subred de red para su máquina virtual remota. Proporcione un nombre de subred, como remotesubnet, junto con un prefijo de dirección dentro del intervalo de la red virtual, como 10.0.2.0/24. También se conectan a la subred los NSG que se crearon en el paso 3, como remotensg:

```
az network vnet subnet create \
    --resource-group azremolchapter5 \
    --vnet-name vnetmol \
    --name remotesubnet \
    --address-prefix 10.0.2.0/24 \
    --network-security-group remotensg
```

Apenas se necesitan tres comandos para crear una subred, una NSG y una regla. ¿Empieza a ver el poder de la CLI de Azure? Azure PowerShell es igual de poderoso, así que no sienta que debe crear todos los recursos en Azure Portal. A medida que avanza en el libro, utilizará la CLI de Azure en lugar del portal en la mayoría de los casos.

5.3.2 Creación de VM

Ahora que tenemos todos los componentes de red, está listo para crear dos VM. Una VM se crea en la subred que permite el tráfico HTTP para poder instalar un servidor web. La otra VM se crea en la subred que permite SSH para que tenga un servidor de salto y así asegurar aún más su entorno de aplicación y comenzar a replicar una implementación de producción. La figura 5.5 le recuerda lo que está compilando.

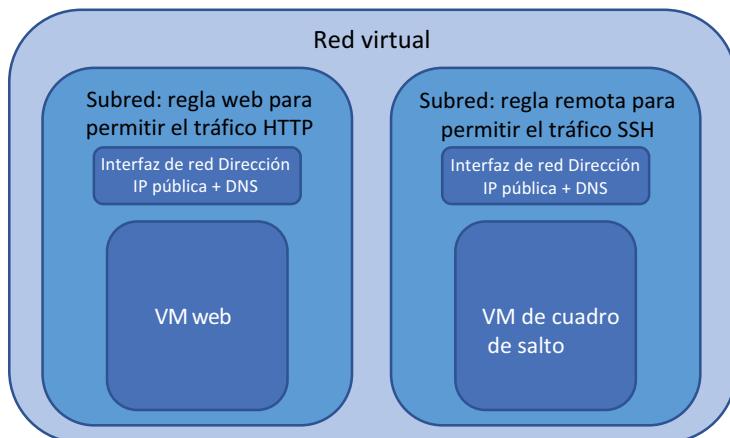


Figura 5.5 Está uniendo dos subredes, NSG, reglas, interfaces de red y VM. Este ejemplo se parece a una implementación lista para producción en la que una VM ejecuta el servidor web y está abierta al tráfico público, y otra VM en una subred separada que se utiliza para conexiones remotas con el resto del entorno de aplicación.

Al crear una VM, puede proporcionar la interfaz de red virtual que creó antes. Si no especificó este recurso de red, la CLI de Azure crea una red, subred y NIC virtuales automáticamente usando los valores predeterminados integrados. Esa opción es genial

para crear rápidamente una VM; pero le recomendamos seguir el principio de utilizar recursos de red de larga duración que otro equipo pueda administrar y en el que creará sus VM.

Pruébelo ahora

Complete los siguientes pasos para usar la CLI de Azure para crear su VM de servidor web y de servidor de salto:

- 1 Cree la primera VM para su servidor web y escriba un nombre, como webvm. Conecte la interfaz de red, como webvnic, e ingrese una imagen, como UbuntuLTS. Escriba un nombre de usuario, como azuremol. El paso final, --generate-ssh-keys, agrega a la VM las claves SSH que creó en el capítulo 2:

```
az vm create \
    --resource-group azuremolchapter5 \
    --name webvm \
    --nics webvnic \
    --image UbuntuLTS \
    --size Standard_B1ms \
    --admin-username azuremol \
    --generate-ssh-keys
```

- 2 Cree la segunda VM para el servidor de salto. En este ejemplo se muestra cómo se puede utilizar una subred y NSG existentes, y permitir que la CLI de Azure cree la interfaz de red y realice las conexiones apropiadas. Usted crea una dirección IP pública, como remotepublicip, como parte de este comando:

```
az vm create \
    --resource-group azuremolchapter5 \
    --name remotevm \
    --vnet-name vnetmol \
    --subnet remotesubnet \
    --nsg remotensg \
    --public-ip-address remotepublicip \
    --image UbuntuLTS \
    --size Standard_B1ms \
    --admin-username azuremol \
    --generate-ssh-keys
```

El resultado de ambos comandos muestra la dirección IP pública. Anote estas direcciones IP. En el siguiente ejercicio, si trata de SSH a su primera VM para el servidor web, falla. ¿Por qué? Usted puede SSH a la VM remota porque creó una regla de NSG para permitir solo el tráfico HTTP a la VM web.

5.3.3 Uso del agente SSH para conectarse a las VM

Permítame presentarle un acto de magia con SSH que le permite usar el servidor de salto correctamente y conectarse a la VM web sobre la red virtual Azure: se llama *agente SSH*. Este agente solo se aplica a las VM Linux, así que si trabaja de manera principal con las VM Windows y las conexiones de escritorio remoto, no se preocupe si SSH es totalmente nuevo. Si configura el servidor adecuadamente, puede crear conexiones RDP a las VM de Windows desde el servidor de salto con las credenciales remotas locales o con credenciales de dominio.

Un agente SSH puede almacenar sus claves SSH y reenviarlas cuando sea necesario. En el capítulo 2, cuando creó un par de claves públicas SSH, hablamos sobre las claves pública y privada. La clave privada es algo que se queda en su equipo. Solo la clave pública se copia en las VM remotas. Aunque la clave pública se agregó a ambas VM que creó, no puede simplemente SSH a su servidor de salto y, a continuación, SSH a la VM web. ¿Por qué? Porque ese servidor de salto no tiene una copia de su clave privada. Cuando intenta hacer la conexión SSH desde el servidor de salto, no tiene ninguna clave privada para emparejar con la clave pública de la VM web para autenticarse.

La clave privada es algo que se debe proteger, así que no debe tomar el camino fácil y copiar la clave privada del servidor de salto. Cualquier otro usuario que acceda al servidor de salto podría potencialmente obtener una copia de su clave privada y luego hacerse pasar por usted en cualquier lugar donde se utilice la clave. Aquí es donde el agente SSH entra en juego.

Si ejecuta el agente SSH en la sesión de Cloud Shell, puede agregar sus claves SSH a la sesión. Para establecer la conexión SSH al servidor de salto, especifique el uso de este agente para abrir su sesión. Esta técnica le permite transferir eficazmente su clave privada para usarla desde el servidor de salto, sin nunca tener que copiar la clave privada. Cuando aplica SSH desde el servidor de salto a la VM web, el agente SSH pasa su clave privada a través del servidor de salto y le permite autenticarlo.

Pruébelo ahora

Complete los siguientes pasos para utilizar SSH con la VM del servidor de salto:

- 1 En Cloud Shell, inicie el agente SSH de la siguiente manera:

```
eval $(ssh-agent)
```

- 2 Agregue la clave SSH que creó en el capítulo 2 al agente de la siguiente manera:

```
ssh-add
```

- 3 Aplique SSH a su VM de servidor de salto. Especifique el uso del agente SSH con el parámetro `-A`. Ingrese su propia dirección IP pública que obtuvo como resultado cuando creó la VM de servidor de salto:

```
ssh -A azuremol@<publicIpAddress>
```

- 4 Esta es la primera vez que ha creado una conexión SSH a la VM de servidor de salto, así que acepte la notificación para conectarse con las claves SSH.

- 5 ¿Recuerda que creó una asignación de dirección IP privada estática para la VM web en la sección 5.1.2? Esta dirección estática hace que sea mucho más fácil aplicar SSH a ella. Aplique SSH a la VM web de la siguiente manera:

```
ssh 10.0.1.4
```

- 6 Acepte la notificación para continuar la conexión SSH. El agente SSH ha tunelizado su clave SSH privada a través del servidor de salto y le permite conectarse correctamente a la VM web. ¿Y ahora qué? Bueno, tiene un laboratorio para ver este trabajo.

5.4 **Laboratorio: Instalación y prueba del servidor web LAMP**

Ya hizo el trabajo difícil del capítulo. Este laboratorio rápido refuerza la forma de instalar un servidor web y le permite ver en acción la regla de NSG en su VM:

- 1 *Instale un servidor web básico de Linux.* Recuerde el capítulo 2 cuando creó una conexión SSH a la VM y luego instaló el paquete de servidor web LAMP con apt. Desde la conexión SSH a su VM web creada en la sección 5.3.2, instale y configure la pila web LAMP predeterminada.
- 2 *Busque el sitio web predeterminado.* Cuando la pila web LAMP esté instalada, abra su navegador web y vaya a la etiqueta de nombre DNS que ingresó al crear una dirección IP pública en la sección 5.1.3. En el ejemplo, utilicé azuremol.eastus.cloud-app.azure.com. También puede utilizar la dirección IP pública que obtuvo cuando creó la VM web. Recuerde: esa dirección IP pública es distinta de la VM de servidor de salto a la que accedería a través de SSH.

Parte 2

Alta disponibilidad y escalabilidad



Es hora de divertirnos! Ahora que entiende los recursos básicos en Azure, puede adentrarse en áreas como la redundancia, el equilibrio de carga y la distribución geográfica de las aplicaciones. Esta parte es donde las cosas se vuelven entretenidas y los temas que aprendió deben comenzar a mostrar soluciones y prácticas recomendadas que puede utilizar en las implementaciones del mundo real. Azure tiene algunas funciones impresionantes para replicar datos de forma global, distribuir el tráfico de los clientes a la instancia más cercana de su aplicación y escalar automáticamente en función de la demanda. Estas funciones son el poder de la informática en la nube y donde tiene la oportunidad de aportar verdadero valor a su trabajo.

Azure Resource Manager

La mayoría de los días, desea pasar el menor tiempo posible dedicado a la forma de implementar un entorno de aplicación y dedicarse a la implementación real. En muchos entornos de TI, se están comenzando a ver equipos de desarrollo y operaciones que colaboran y trabajan estrechamente, y se escucha el término de moda *DevOps* en muchas conferencias y blogs.

No hay nada inherentemente nuevo o innovador sobre la cultura DevOps, pero es común que los diferentes equipos no trabajen con conjunto como debieran hacerlo. Las herramientas modernas han impulsado el movimiento DevOps, con soluciones de integración continua y entrega continua (CI/CD) capaces de automatizar toda la implementación de entornos de aplicaciones basándose en un único código comprobado por un desarrollador. El equipo de operaciones suele ser el que compila y mantiene estos procesos CI/CD, haciendo posible pruebas e integraciones más rápidas de las actualizaciones de aplicaciones para los desarrolladores.

El modelo de implementación de Azure Resource Manager es fundamental para la forma en que se compilán y ejecutan los recursos, aunque probablemente todavía no se haya dado cuenta. Resource Manager es un método para compilar e implementar recursos, tanto como los procesos de automatización y las plantillas que impulsan esas implementaciones. En este capítulo, aprenderá a utilizar las funciones de Resource Manager, como controles de acceso y bloqueos, implementaciones de plantillas coherentes e implementaciones de múltiples niveles automatizadas.

6.1 *El enfoque de Azure Resource Manager*

Cuando creó una VM o una aplicación web en capítulos anteriores, primero se creó un grupo de recursos como la construcción central para contener todos sus recursos. Un grupo de recursos es central para todos los recursos: una VM, una aplicación web, una red virtual o una tabla de almacenamiento no pueden existir fuera de un grupo de recursos. Pero el grupo de recursos es más que un simple

lugar para organizar sus recursos, mucho más. En esta sección se echa un vistazo al modelo de Azure Resource Manager subyacente y se muestra por qué es importante para compilar y ejecutar aplicaciones.

6.1.1 Diseño alrededor del ciclo de vida de las aplicaciones

Idealmente, cuando se compila una aplicación es sumamente importante mantenerla. Por lo general, tiene actualizaciones que desarrollar e implementar, nuevos paquetes que instalar, nuevas VM que agregar y ranuras de implementación de aplicaciones web adicionales que crear. Es posible que deba realizar cambios en la configuración de la red virtual y en las direcciones IP. En capítulos anteriores mencioné que sus redes virtuales en Azure pueden ser administradas por un equipo diferente. Empiece a pensar en cómo ejecutar en una escala global, grande, y en términos del ciclo de duración y administración de las aplicaciones.

Tiene un par de enfoques principales para agrupar los recursos en Azure:

- *Todos los recursos de una aplicación determinada en el mismo grupo de recursos:* como se muestra en la figura 6.1, este enfoque funciona bien para aplicaciones más pequeñas y para entornos de desarrollo y pruebas. Si no necesita compartir espacios de red grandes y puede administrar individualmente el almacenamiento, puede crear todos los recursos en un lugar y, a continuación, administrar las actualizaciones y los cambios de configuración mediante una sola operación.

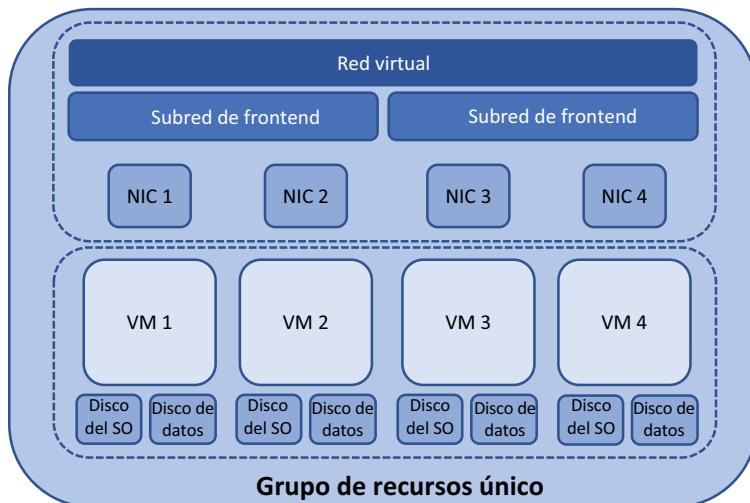


Figura 6.1 Una forma de compilar una aplicación en Azure es crear todos los recursos relacionados con esa implementación de aplicaciones en el mismo grupo de recursos y administrarlos como una entidad.

- *Recursos afines agrupados por función en el mismo grupo de recursos:* como se muestra en la figura 6.2, este enfoque suele ser más común en aplicaciones y entornos más grandes. La aplicación puede existir en un grupo de recursos solo con las VM y los componentes de aplicación compatibles. Los recursos de red virtual y las direcciones IP pueden existir en un grupo de recursos diferente, protegidos y administrados por un grupo diferente de ingenieros.

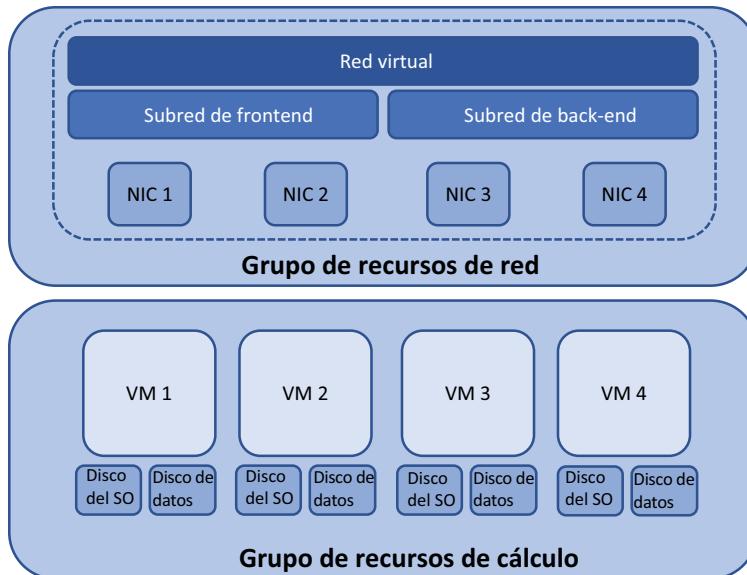


Figura 6.2 Un enfoque alternativo es crear y agrupar recursos basándose en su función. Un ejemplo común es que todos los recursos de red principales se encuentran en un grupo de recursos distinto de los procesos de la aplicación principal. Las VM del grupo de procesos pueden tener acceso a los recursos de red del grupo separado, pero los dos conjuntos de recursos se pueden administrar y proteger de forma independiente.

¿Por qué hay diferentes enfoques? La respuesta no se relaciona completamente con asegurar el trabajo ni con los silos con que algunos equipos prefieren trabajar. Se trata de cómo necesita administrar los recursos subyacentes. En entornos y aplicaciones más pequeños donde todos los recursos existen en el mismo grupo de recursos, usted es responsable de todo en ese entorno. Este enfoque también es adecuado para desarrollo y los entornos de prueba donde todo está empaquetado junto. Los cambios que realice a la red virtual solo afectarán a su aplicación y grupo de recursos.

La verdad es que las redes no cambian de manera frecuente. Los rangos de direcciones suelen estar bien definidos y planificados, de modo que pueden coexistir a través de Azure y oficinas en todo el mundo. Lógicamente, a menudo tiene sentido colocar los componentes de red en su propio grupo de recursos. La red se administra separadamente de la aplicación y el almacenamiento puede administrarse y actualizarse separadamente de la misma manera. No hay nada inherentemente malo en dividir los recursos de esta forma, siempre y cuando el personal de TI no se quede atascado en una mentalidad de silo, ya que resulta en falta de cooperación.

Para sus aplicaciones, la división de recursos también puede ser una ventaja, ya que tiene la libertad suficiente para hacer los cambios y actualizaciones que deseé. Precisamente porque no tiene los componentes de red en el grupo de recursos, no necesita preocuparse por ellos cuando realice actualizaciones de las aplicaciones.

6.1.2 Protección y control de recursos

Cada recurso puede tener diferentes permisos de seguridad aplicados. Estas directivas definen quién puede hacer qué. Piénselo: ¿desearía que un alumno en práctica reinicie su aplicación web o elimine los discos de datos de la VM? ¿Y cree que sus amigos del equipo de redes quieren que usted cree una nueva subred de red virtual? Probablemente no. En Azure, hay cuatro roles principales que puede asignar a los recursos, similar a los permisos de archivo:

- *Propietario*: control completo, básicamente un administrador
- *Colaborador*: administración completa del recurso, excepto para realizar cambios en las asignaciones de roles y seguridad
- *Lector*: capacidad de ver toda la información del recurso, pero no puede realizar cambios.
- *Administrador de acceso de usuarios*: capacidad de asignar o eliminar el acceso a los recursos.

El control de acceso basado en roles (RBAC) es una función principal de Azure Resources que se integra automáticamente con las cuentas de usuario en sus suscripciones. Piense en los permisos de archivo en su equipo normal. Los permisos de archivo básicos son leer, escribir y ejecutar. Cuando se combinan estos permisos, se pueden crear diferentes conjuntos de permisos para cada usuario o grupo del equipo. Al trabajar con recursos compartidos de archivos de red, los permisos son herramientas comunes para controlar el acceso. RBAC en Azure trabaja en las mismas líneas para controlar el acceso a los recursos, al igual que los permisos de archivo en su equipo local o en red compartida; (figura 6.3).

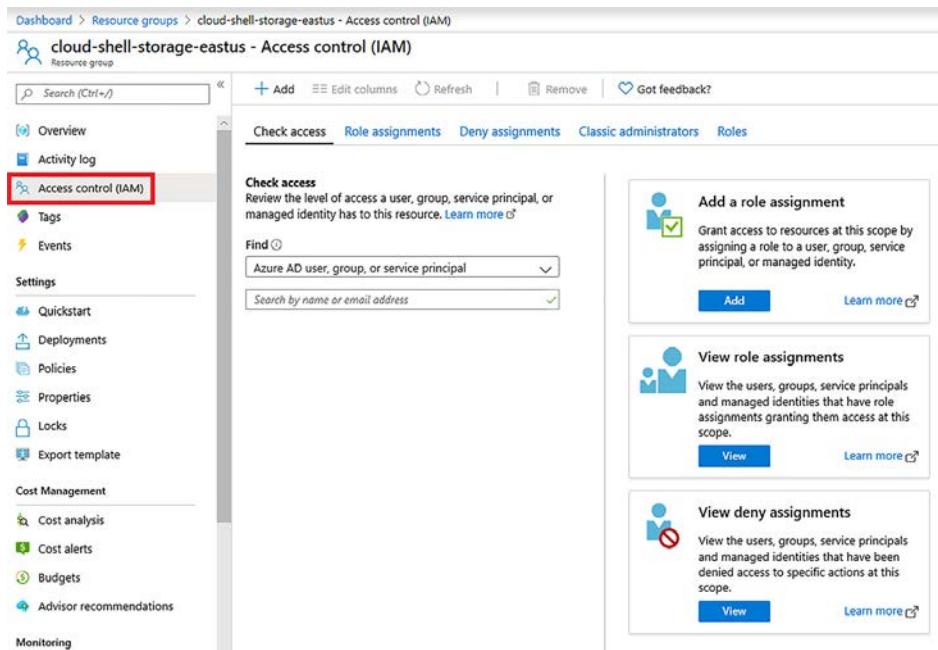


Figura 6.3 El control de acceso para cada recurso Azure enumera cuáles son las asignaciones actuales. Puede agregar asignaciones o seleccionar Roles para ver información sobre los conjuntos de permisos disponibles.

Pruébelo ahora

Abra Azure Portal en un navegador web y, a continuación, seleccione cualquier recurso que tenga, como cloud-shell-storage. Elija el botón Control de acceso (IAM), como se muestra en la figura 6.3. Revise las asignaciones de roles actuales. Vea cómo agregar una asignación de roles y explore todas las asignaciones disponibles. El ícono de información de cada rol muestra qué permisos se asignan.

A medida que explora las roles disponibles, puede observar varios roles específicos de los recursos, entre ellos:

- Colaborador de máquina virtual
- Colaborador de sitio web
- Colaborador de red

¿Puede adivinar qué significan esos roles? Toman la función de Colaborador de la plataforma principal y lo aplican a un tipo de recurso específico. Este caso de uso se remonta al concepto de manejar los recursos afines. Es posible que se le haya asignado el rol de colaborador de máquina virtual o de sitio web. A continuación, cualquier VM o aplicaciones web creadas en ese grupo de recursos estarán disponibles para que usted las administre. Pero no puede administrar los recursos de red, que pueden estar en un grupo de recursos completamente diferente.

6.1.3 Protección de recursos con bloqueos

El enfoque basado en permisos de RBAC es genial para limitar quién puede acceder a qué. Pero los errores siempre ocurren. Hay una razón por la que normalmente no inicia sesión en un servidor como usuario con permisos administrativos o raíz. Una tecla o clic incorrecto, y podría eliminar recursos por equivocación. Aunque tenga copias de seguridad (las tiene, ¿verdad? ¿y las prueba periódicamente?), es un proceso que requiere mucho tiempo y que puede significar pérdida de productividad o ingresos para el negocio. En el capítulo 13, aprenderá más sobre las formas en que los servicios de copia de seguridad, recuperación y replicación de Azure protegen sus datos.

Otra función incorporada al modelo de Resource Manager son los bloqueos de recursos. Cada recurso puede tener un bloqueo aplicado que lo limita para acceso solo de lectura o impide las operaciones de eliminación. El bloqueo de eliminación es particularmente útil, ya que puede ser demasiado fácil eliminar el grupo de recursos incorrecto. Cuando se inicia una operación de eliminación, no hay vuelta atrás ni se puede cancelar la operación después de que la plataforma Azure ha aceptado su solicitud.

Para las cargas de trabajo de producción, le sugiero que implemente bloqueos en los recursos principales para evitar que se eliminen. Estos bloqueos son solo en los niveles de recursos y plataformas Azure, y no para los datos dentro de sus recursos. Por ejemplo, puede eliminar archivos dentro de una VM o eliminar la tabla de una base de datos. Los bloqueos de recursos Azure solo se aplicarían si intentó eliminar toda la base de datos de VM o Azure SQL. La primera vez que se inicie un bloqueo y evite la eliminación del grupo de recursos incorrecto, ¡me lo agradecerá!

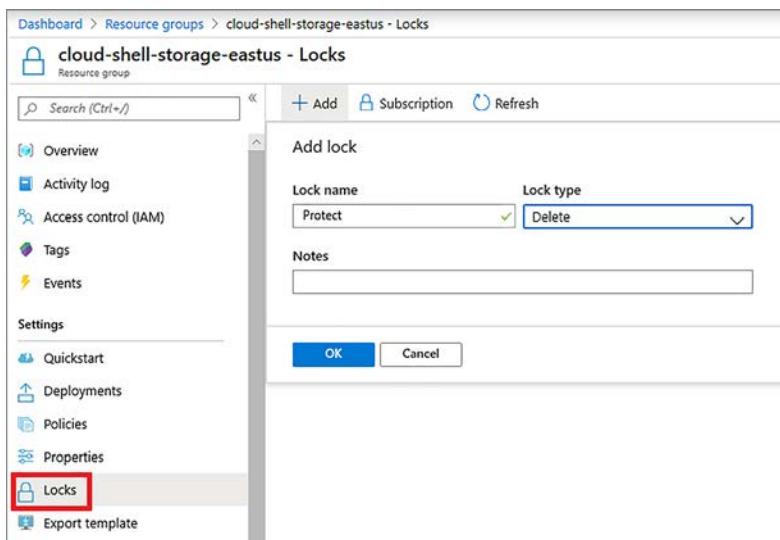


Figura 6.4 Cree un bloqueo de recursos en Azure Portal.

Pruébelo ahora

Complete los siguientes pasos para ver los bloqueos de recursos Azure en acción, como se muestra en la figura 6.4:

- 1 Abra Azure Portal en un navegador web y, a continuación, seleccione cualquier grupo de recursos que tenga, como cloud-shell-storage.
- 2 Seleccione Bloqueos en el lado izquierdo del portal.
- 3 Escriba un Nombre de bloqueo, como Proteger; elija Eliminar del menú desplegable Tipo de bloqueo; elija Aceptar. El nuevo bloqueo aparece en la lista.
- 4 Seleccione Descripción general para el grupo de recursos y, a continuación, intente eliminar el grupo de recursos. Debe escribir el nombre del grupo de recursos para confirmar que desea eliminarlo (lo que es además un buen aviso mental para asegurarse de que va a eliminar el recurso correcto).
- 5 Cuando elija el botón Eliminar, revise el mensaje de error que se muestra para ver cómo el bloqueo impidió que Azure eliminara el recurso.

6.1.4 Administración y agrupación de recursos con etiquetas

Una última función del modelo Azure Resource Manager que queremos mencionar son las *etiquetas*. No hay nada nuevo o especial acerca de cómo etiquetar recursos en Azure, pero este concepto de administración suele pasarse por alto. Puede aplicar etiquetas a un recurso de Azure que describa propiedades como la aplicación de la que forma parte, el departamento responsable de la misma, o si se trata de un recurso de desarrollo o de producción.

A continuación, puede identificar recursos basándose en las etiquetas para aplicar bloqueos o roles RBAC, o informar sobre los costos y el consumo de los recursos. Las etiquetas no son exclusivas de un grupo de recursos y pueden reutilizarse en su suscripción. Se pueden aplicar hasta 50 etiquetas a un único recurso, por lo que tiene mucha flexibilidad en la forma de etiquetar y luego filtrar los recursos etiquetados.

Pruébelo ahora

Complete los siguientes pasos para ver las etiquetas de recursos de Azure en acción:

- 1 Abra Azure Portal en un navegador web y, a continuación, seleccione cualquier recurso que tenga, como cloud-shell-storage. Aunque puede etiquetar un grupo de recursos por sí mismo, no elija un grupo de recursos para este ejercicio.
- 2 Con su recurso seleccionado, elija el botón Etiquetas, como se muestra en la figura 6.5.
- 3 Escriba un Nombre, como workload y un Valor, como development.
- 4 Seleccione Guardar.
- 5 Abra Cloud Shell.
- 6 Para filtrar los recursos basados en etiquetas, utilice la lista de recursos az con el parámetro --tag. Utilice su propio nombre y valor de la siguiente manera:

```
az resource list --tag workload=development
```

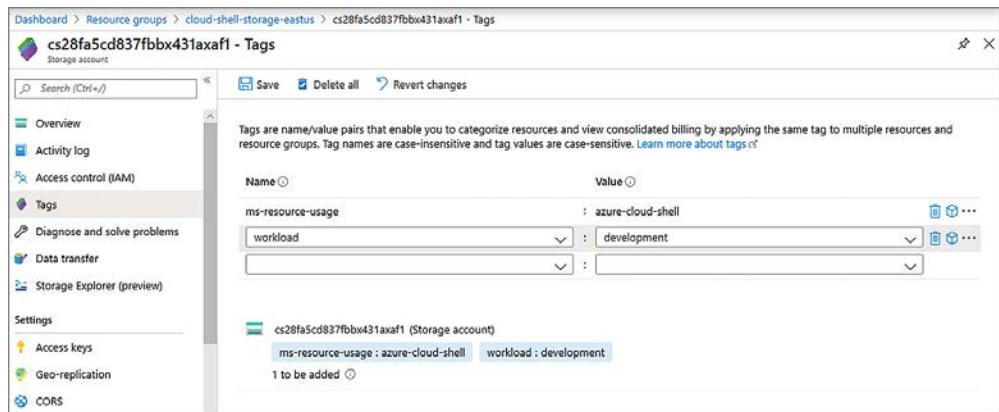


Figura 6.5 Puede crear hasta 50 etiquetas name:value para cada recurso o grupo de recursos de Azure.

6.2 Plantillas de Azure Resource Manager

Hasta ahora, ha creado un pequeño número de recursos Azure a la vez. Para ello, utilizó Azure Portal o la CLI de Azure. Aunque no le hemos mostrado Azure PowerShell, hablamos acerca de este en el primer capítulo y está disponible en Cloud Shell. Tal vez lo haya probado sin mí. ¡Mejor aún! Como mencioné en el capítulo 1,

Azure tiene herramientas que le permiten elegir lo más cómodo para usted y el entorno en el que trabaja.

La desventaja de utilizar el portal o los comandos CLI o PowerShell es que debe hacer clic en un montón de botones del navegador web o escribir líneas de comandos para compilar el entorno de la aplicación. Puede crear scripts para hacer todas estas cosas, pero luego debe crear la lógica para controlar la forma de crear varios recursos al mismo tiempo o el orden en el que crearlos.

Un script que reúne los comandos de la CLI de Azure o PowerShell comienza a moverse en la dirección correcta en términos de cómo debe compilar e implementar entornos de aplicaciones, no solo en Azure, sino en cualquier plataforma. Hay una tendencia hacia la infraestructura como código (IaC), que no es nada nuevo si ha estado en el mundo de TI durante algún tiempo. Básicamente, significa que no confía en un humano para escribir comandos y seguir un conjunto de pasos; en lugar de ello, crea su infraestructura mediante programación a partir de un conjunto de instrucciones. Las implementaciones manuales introducen un elemento humano que a menudo puede conducir a pequeñas configuraciones erróneas y diferencias en las VM finales, como se muestra en la figura 6.6.

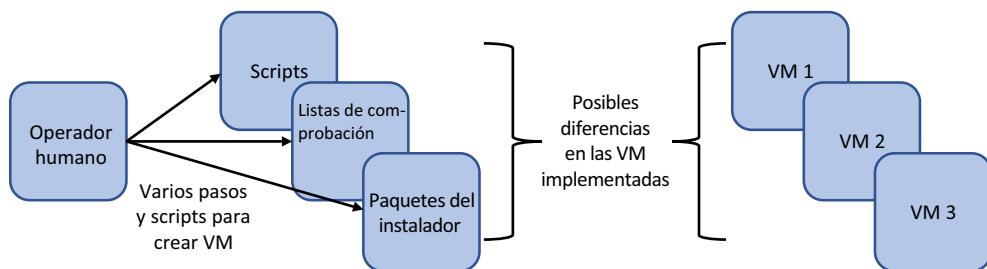


Figura 6.6 Los seres humanos cometen errores, como tipar mal un comando o saltarse un paso en una implementación, lo que puede generar una VM con un resultado ligeramente diferente. La automatización se utiliza a menudo para quitar el operador humano de la ecuación y así poder crear implementaciones uniformes e idénticas cada vez.

Incluso cuando tiene scripts, todavía necesita que alguien los escriba, mantenga y actualice a medida que se liberan las nuevas versiones de los módulos CLI de Azure o PowerShell. Sí, a veces hay cambios importantes en las herramientas para acomodar las nuevas funciones, aunque son raras.

6.2.1 Creación y uso de plantillas

Las plantillas de Resource Manager pueden ayudar a reducir el error humano y la dependencia de scripts escritos manualmente. Las plantillas se escriben en JavaScript Object Notation (JSON), un enfoque abierto, estándar y multiplataforma que permite editarlas en un editor de texto básico. Con las plantillas, puede crear implementaciones uniformes y reproducibles que minimicen los errores. Otra función incorporada de las plantillas es que la plataforma comprende las dependencias y puede crear recursos en paralelo cuando sea posible para acelerar el tiempo de implementación. Por ejemplo, si crea tres VM, no es necesario esperar a que la primera VM finalice la implementación antes de crear la segunda; Resource Manager puede crear las tres VM al mismo tiempo.

Como ejemplo de las dependencias, si crea una NIC virtual, es necesario conectarla a una subred. Lógicamente, la subred debe existir antes de poder crear la NIC virtual, y

la subred debe formar parte de una red virtual, por lo que esa red debe crearse antes de la subred. La figura 6.7 muestra la cadena de dependencias en acción. Si intenta escribir un script usted mismo, debe planificar cuidadosamente el orden en que se crean los recursos, e incluso después, debe crear una lógica para saber cuando los recursos primarios estén listos y pueda pasar a los recursos dependientes.

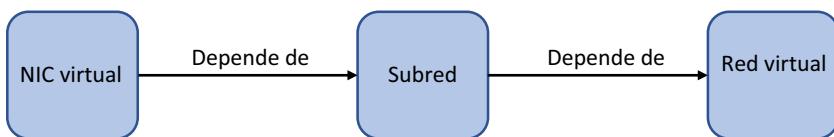


Figura 6.7 Azure Resource Manager maneja las dependencias automáticamente. La plataforma conoce el orden en el que crear recursos y reconoce el estado de cada uno sin el uso de lógica manuscrita y bucles como los que debe utilizar en sus propios scripts.

¿Quiere saber algo asombroso? Ya utilizó plantillas de Resource Manager en el capítulo 2 y en la primera VM que creó. Al crear una VM en el portal o en la CLI de Azure, a la vez se crea y se implementa una plantilla mediante programación. ¿Por qué? Bueno, ¿para qué reinventar la rueda y pasar por el proceso de desarrollar toda esa lógica para las implementaciones? ¡Deje que Azure Resource Manager lo haga automáticamente!

Así es cómo se ve una sección de una plantilla de Resource Manager. El siguiente listado muestra la sección que crea una dirección IP pública, tal como en ejemplos anteriores cuando creó una VM.

Listado 6.1 Creación de una dirección IP pública en una plantilla de Resource Manager

```

{
  "apiVersion": "2019-04-01",
  "type": "Microsoft.Network/publicIPAddresses",
  "name": "publicip",
  "location": "eastus",
  "properties": {
    "publicIPAllocationMethod": "dynamic",
    "dnsSettings": {
      "domainNameLabel": "azuremol"
    }
  }
},
  
```

Incluso si JSON es nuevo para usted, está escrito en un formato bastante legible para el ser humano. Primero define un tipo de recurso, en este ejemplo, Microsoft.Network/publicIPAddresses. Luego, proporciona un nombre, como publicip y una ubicación, como eastus. Finalmente, define el método de asignación dynamic en este ejemplo, y un nombre de etiqueta DNS, como azuremol. Estos son los mismos parámetros que proporcionó cuando usó Azure Portal o la CLI de Azure. Si usa PowerShell, adivine qué... Se solicitan los mismos parámetros.

La diferencia con la plantilla es que no tenía que escribir ninguna información. Toda la información estaba en el código. Puede estar pensando: "Genial, pero ¿qué

pasa si quiero usar nombres diferentes cada vez?" Al igual que con un script, puede asignarlos dinámicamente usando parámetros y variables:

- Los *parámetros* son valores que se le solicitan. A menudo se utilizan para las credenciales de usuario, el nombre de la VM y la etiqueta de nombre DNS.
- Se *puede asignar* de forma previa un valor a las variables, pero también se ajustan cada vez que se implementa la plantilla, como el tamaño de VM o el nombre de red virtual.

Pruébelo ahora

Para ver una plantilla completa de Resource Manager, abra un navegador web en el repositorio de GitHub en <http://mng.bz/QyWv>.

6.2.2 Creación de varios tipos de recursos

A medida que compile sus plantillas, intente ir pensando cómo podría necesitar ampliar sus aplicaciones en el futuro. Es posible que solo necesite una VM única cuando implemente por primera vez la aplicación, pero a medida que aumente su demanda, podría tener que crear instancias adicionales.

En una implementación tradicional con scripts, se crean bucles `for` o `while` para crear varios tipos de recursos. ¡Resource Manager tiene esta funcionalidad incorporada! Hay más de 50 tipos de funciones en Resource Manager, al igual que en la mayoría de los lenguajes de programación y lenguajes de scripts. Dentro de funciones comunes de Resource Manager se incluyen `length`, `equals` or `trim`. Usted controla el número de instancias que se crearán con la función `copy`.

Cuando utilice la función `copy`, Resource Manager creará el número de recursos que especifique. Cada vez que Resource Manager reitera la operación Crear, habrá un valor numérico disponible para que nombre recursos de manera secuencial. Puede acceder a este valor con la función `copyIndex()`. En el ejemplo del listado 6.1 se creó una sola dirección IP pública. En el ejemplo del listado 6.2 se utiliza el mismo tipo de proveedor de recursos `Microsoft.Network/publicIPAddresses`, pero se crean dos direcciones IP públicas. Utilice `copy` para definir cuántas direcciones desea crear y `copyIndex()` para nombrar las direcciones secuencialmente.

Listado 6.2 Creación de varias direcciones IP públicas con `copy`

```
{
    "apiVersion": "2019-04-01",
    "type": "Microsoft.Network/publicIPAddresses",
    "name": "[concat('publicip', copyIndex())]",
    "copy": {
        "count": 2
    },
    "location": "eastus",
    "properties": {
        "publicIPAllocationMethod": "dynamic",
    }
},
```

También se utiliza la función concat para combinar el nombre de la dirección IP pública y el valor numérico de cada instancia que se crea. Una vez implementada esta plantilla, sus dos direcciones IP públicas se denominan publicip0 y publicip1. Estos nombres no son muy descriptivo, pero este ejemplo básico muestra el concepto de cómo se puede utilizar una convención de numeración a medida que se crean varios recursos de la función copy.

6.2.3 Herramientas para compilar sus propias plantillas

Digamos las cosas como son: si bien las plantillas de Resource Manager son claras y una de las principales formas en las que recomendamos compilar e implementar aplicaciones en Azure, necesitará escribir las plantillas de todas maneras. Un par de herramientas diferentes simplifican esta tarea, y hay disponibles cientos de plantillas de ejemplo de Microsoft y de terceros. De hecho, una de las mejores maneras de aprender cómo crear y utilizar plantillas es examinar las plantillas de inicio rápido que Microsoft pone a su disposición en su repositorio de ejemplos en <https://github.com/Azure/azure-quickstart-templates>.

Si quiere subirse las mangas y empezar a escribir sus propias plantillas, le recomendamos dos herramientas principales. La primera es Visual Studio Code, un editor multiplataforma, gratuito y open source (<https://code.visualstudio.com>). Junto con algunas funcionalidades incorporadas como control de código fuente e integración de GitHub, hay extensiones disponibles que pueden compilar automáticamente las diferentes secciones, o proveedores, para que los recursos construyan una plantilla, como se muestra en la figura 6.8. Si descarga e instala VS Code, elija a Ver > Extensiones y luego busque *Azure*.

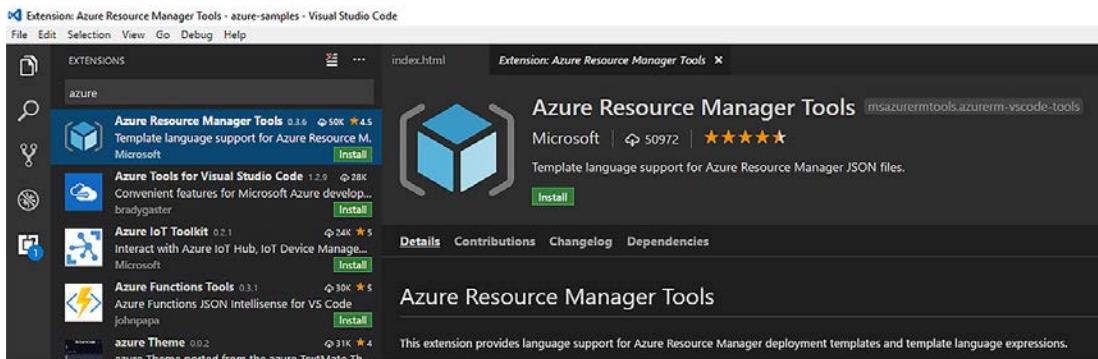


Figura 6.8 Hay muchas extensiones disponibles en Visual Studio Code para mejorar y optimizar la forma de crear y utilizar plantillas de Azure Resource Manager.

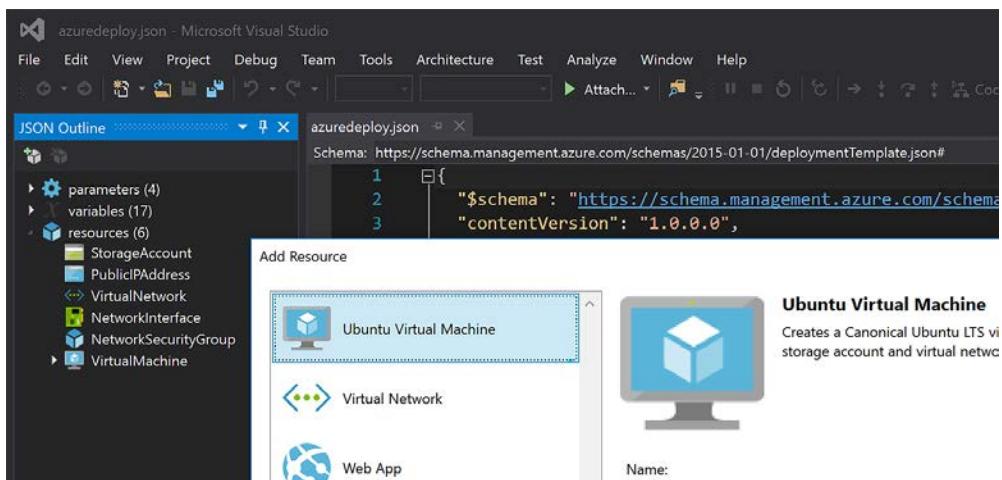


Figura 6.9 Con Visual Studio, puede crear plantillas gráficamente y explorar recursos JSON.

Una forma más gráfica de crear plantillas de Azure Resource Manager es utilizar el editor de Visual Studio completo, como se muestra en la figura 6.9. Hay versiones para Windows y macOS, pero se necesita una licencia distinta para usar el editor. Existe una edición comunitaria, pero tenga cuidado si compila plantillas dentro de su empresa: normalmente necesita una versión con licencia. Consulte a sus expertos en licencias, porque Visual Studio está dirigido a los desarrolladores de aplicaciones.

Por supuesto, puede utilizar un editor de texto básico. Parte de la razón por la que las plantillas de Azure Resource Manager están escritas en JSON es que elimina la necesidad de cualquier herramienta especial. Hay una curva de aprendizaje para trabajar con JSON, motivo por el que le recomendamos que explore las plantillas de inicio rápido en el repositorio de ejemplos de Azure. Tenga mucho cuidado con la sangría, las comas al final, el uso de paréntesis, los corchetes y las llaves.

Vida en Marte

Hay herramientas de terceros y otras formas de usar plantillas en Azure. Hashicorp proporciona muchas herramientas y soluciones open source para informática en la nube, una de ellas es Terraform. Con Terraform, se definen todos los recursos que se desean incorporar de la misma manera que lo hace una plantilla de Azure Resource Manager Azure nativa. También puede definir dependencias y utilizar variables. La diferencia es que Terraform es técnicamente un proveedor de servicios cruzados. Se puede utilizar el mismo enfoque de construcciones y plantillas en Azure, Google Cloud, AWS y vSphere, por ejemplo. La diferencia son los proveedores que se utilizan para cada recurso.

¿Se trata realmente de enfoque "una plantilla para cualquier proveedor"? Para nada. Terraform es también una aplicación que analiza su plantilla y luego se comunica con el proveedor de nube pertinente, como Azure. Usted tiene cero capacidad de edición (y mucho menos herramientas gráficas) para compilar su plantilla. Elige un editor

y escribe la plantilla a mano. Así que la mejor manera de aprender Terraform es explorar la documentación y plantillas de ejemplo que ofrece.

La razón por la que presento este tema se relaciona al concepto de elección en Azure. Si encuentra que las plantillas de Azure Resource Manager escritas en JSON son demasiado engorrosas, entonces explore un producto como Terraform. Pero no se dé por vencido con las implementaciones de Resource Manager basadas en plantillas. Para lograr implementaciones reproducibles y uniformes a escala, las plantillas son el mejor enfoque, así que busque uno que funcione bien para su caso.

6.2.4 Almacenamiento y uso de plantillas

Así que le encanta la idea de las plantillas de Azure Resource Manager e instaló Visual Studio o Code para escribir sus propias plantillas. Ahora, ¿cómo las almacena e implementa? En el laboratorio de fin del capítulo, implementará una plantilla desde el repositorio de ejemplos de Azure en GitHub. Este repositorio es público, y probablemente no quiera que sus plantillas de aplicación estén a disposición de todo el mundo.

Hay un par de métodos comunes para almacenar las plantillas de Resource Manager de forma privada:

- Uso de un repositorio privado o un recurso compartido de archivos de red dentro de su organización.
- Uso de Azure Storage para almacenar y proteger las plantillas de forma centralizada para su implementación.

No hay manera correcta o incorrecta de almacenar e implementar plantillas, tiene la flexibilidad de utilizar cualquier proceso y herramientas existentes. La ventaja de utilizar un repositorio es que normalmente tiene algún tipo de control de versión para que pueda garantizar implementaciones uniformes y revisar el historial de sus plantillas si fuera necesario. La única limitación es que cuando se implementa la plantilla, es necesario proporcionar las credenciales apropiadas para acceder a la ubicación compartida. Este proceso de autenticación puede variar, como proporcionar un nombre de usuario o token de acceso como parte de la dirección URL a una plantilla de un repositorio o proporcionar un token de firma de acceso compartido (SAS) si utiliza Azure Storage.

Los repositorios públicos como GitHub también son grandes maneras de aprender y compartir. Le sugerimos que mantenga sus plantillas de producción almacenadas en privado, pero si crea una plantilla clara para un entorno de laboratorio o para probar algunas funciones nuevas, puede compartirla en GitHub para devolverle la mano a la comunidad de TI y ayudar a otros que quieran hacer la misma implementación. A medida que comience a compilar sus propias plantillas, asegúrese de comprobar qué plantillas ya existen para que no empiece desde cero y tenga que reinventar la rueda cada vez.

6.3 Laboratorio: Implementación de recursos de Azure desde una plantilla

Toda esta teoría sobre los modelos y enfoques de implementación es fantástica, pero (idealmente) empezará a ver las ventajas y su eficiencia cuando utilice plantillas de verdad:

- 1 Vaya a Ejemplos de inicio rápido de Azure en GitHub (<https://github.com/Azure/azure-quickstart-templates>), y encuentre uno de su interés. Un buen lugar para empezar es una simple VM Linux o Windows.
- 2 Hay botones incorporados en las muestras de GitHub para implementar directamente a Azure. Cuando encuentre una plantilla que le guste, seleccione Implementar en Azure, como se muestra en la figura 6.10, y siga los pasos del portal. El proceso es muy parecido al de la creación de una VM, pero solo se necesitan unas pocas indicaciones para completar los parámetros necesarios. Todos los otros recursos se crean e incorporan.
- 3 El paso final para implementar la plantilla es aceptar el acuerdo de licencia que se muestra y luego seleccionar Comprar. Cuando implementa una plantilla, está creando recursos de Azure, por lo que Comprar significa que acuerda pagar los costos de esos recursos de Azure.

Una de las plantillas básicas, como una VM Linux o Windows simple, cuesta aproximadamente lo mismo que cualquier otra VM que haya creado hasta ahora. Asegúrese de eliminar el grupo de recursos cuando finalice la implementación, así como limpiar después de cualquier otro ejercicio.

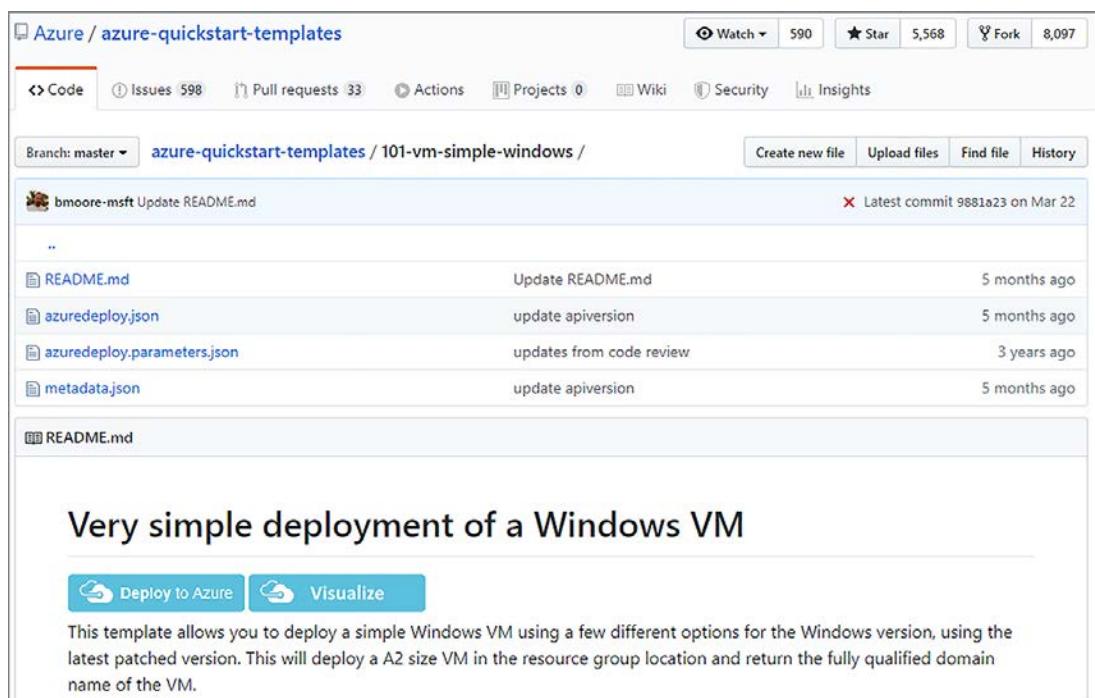


Figura 6.10 Para cada plantilla de Resource Manager en el repositorio de ejemplos de GitHub, hay un botón Implementar en Azure. Si selecciona este botón, se cargará Azure Portal y la plantilla. Se le solicitan algunos parámetros básicos y el resto de la implementación la controla la plantilla.

Parámetros en plantillas

Como vimos en la sección 6.2.1, puede utilizar parámetros y variables en sus plantillas. Recuerde, los parámetros son valores que se solicitan, y las variables son valores dinámicos que se pueden aplicar en una plantilla. Los valores que se le solicitan (parámetros) varían de una plantilla a otra. Por lo tanto, dependiendo de la plantilla de inicio rápido que seleccione, puede que se le solicite uno o dos valores, o puede que tenga que proporcionar siete u ocho.

A medida que diseña sus plantillas, intente prever cómo usted y otros usuarios podrían querer volver a utilizar la plantilla al implementar aplicaciones. Puede proporcionar un valor predeterminado y limitar los valores permitidos. Tenga cuidado con estos valores predeterminados y permisibles, o podría restringir demasiado a los usuarios y forzarlos a crear sus propias plantillas. Cuando sea posible, intente crear plantillas de núcleo reutilizables que tengan suficiente flexibilidad.

- 4 Cuando su plantilla haya sido implementada, vuelva a GitHub y examine el archivo `azure-deploy.json`. Este archivo corresponde a la plantilla de Azure Resource Manager que usó para crear e implementar el ejemplo. Vea si puede entender los diferentes tipos de recursos y configuraciones que se aplicaron. A medida que trabaje con más tipos de recursos y plantillas de Azure, el formato JSON será más fácil de entender. Lo prometo.

Alta disponibilidad y redundancia

No puedo contar el número de veces que algo en TI me ha fallado. El disco duro de mi equipo portátil dejó de funcionar el día antes de una conferencia, una fuente de alimentación humeando en un servidor de correo electrónico y fallas en las interfaces de red en un router de núcleo. Y ni siquiera quiero mencionar las actualizaciones de sistema operativo, controlador y firmware. Estoy seguro de que cualquier persona que trabaje en TI le encantaría compartir historias de horror de las situaciones con las que han tenido que lidiar, especialmente las ocurridas a última hora de la noche o en un momento crucial para la empresa. ¿Sucede alguna vez algo así como un buen fracaso en un buen momento?

Si prevé fallas en TI, aprende a planificar y diseñar sus aplicaciones para que se adapten a los problemas. En este capítulo, aprenderá a utilizar funciones de alta disponibilidad y redundancia de Azure para minimizar los cortes causados por actualizaciones de mantenimiento e interrupciones. Este capítulo crea una base de conocimientos para los próximos dos o tres capítulos a medida que comienza a pasar de una aplicación que se ejecuta en una sola VM o aplicación web, a una que puede escalar y distribuirse globalmente.

7.1 *La necesidad de redundancia*

Si desea que los clientes puedan confiar en usted sus importantes negocios de pizza, debe suministrar aplicaciones a las que puedan acceder cuando las necesiten. La mayoría de los clientes no buscan "horas de atención" en un sitio web, especialmente si trabaja en un entorno global y los clientes están en todo el mundo. ¡Cuando tienen hambre, quieren comer!

La figura 7.1 muestra un ejemplo básico de una aplicación que se ejecuta en una sola VM. Desafortunadamente, esta aplicación crea un punto de falla único. Si esa VM no está disponible, entonces la aplicación no está disponible, y eso conduce a la insatisfacción y hambre del cliente.

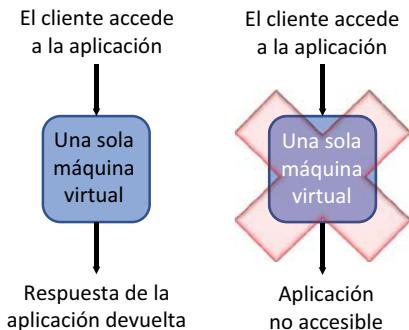


Figura 7.1 Si la aplicación se ejecuta en una sola VM, cualquier interrupción en esa VM hace que la aplicación sea inaccesible. Esto podría significar que los clientes cambiarán de empresa o, al menos, que no estarán satisfechos con el servicio que usted proporciona.

Si conduce un automóvil, es muy probable que cuente con una llanta de repuesto por si sufre un pinchazo. Si utiliza un equipo portátil o una tableta, es muy probable que conecte el dispositivo a un cargador si la batería se agota en medio del trabajo. En casa o en su apartamento, ¿tiene bombillas de repuesto en caso de que se apague alguna de las luces? ¿Y linterna o velas en caso de que haya un apagón?

A la mayoría de la gente le gusta tener algún tipo de plan de redundancia o respaldo, tanto en la vida cotidiana como, especialmente, en TI. Si está listo para cambiar una llanta del automóvil o si tiene una bombilla de repuesto, puede manejar interrupciones y fallas con una interrupción mínima. Si diseña y compila sus aplicaciones con redundancia, proporciona un alto nivel de disponibilidad a sus clientes, lo que minimiza, o incluso oculta, cualquier interrupción que sufra la aplicación. Todos los centros de datos de Azure están construidos para alta disponibilidad. Suministro eléctrico de respaldo, varias conexiones de red y matrices de almacenamiento con discos de repuesto son solo algunos de los conceptos básicos de redundancia que Azure proporciona y administra para usted. Sin embargo, toda la redundancia que ofrece Azure no será suficiente si ejecuta su aplicación en una sola VM. Para brindarle flexibilidad y control para que su aplicación esté altamente disponible, existen dos funciones principales para las cargas de trabajo de IaaS:

- *Zona de disponibilidad*: permite distribuir VM en segmentos físicamente aislados de una región de Azure para maximizar aún más la redundancia de la aplicación. Las zonas también pueden proporcionar alta disponibilidad a recursos de red como direcciones IP públicas y equilibradores de carga.
- *Conjunto de disponibilidad*: permite agrupar las VM de forma lógica para distribuirlas a través de un único centro de datos de Azure y minimizar los cortes producto de actualizaciones de mantenimiento o interrupciones.

Para la mayoría de las implementaciones de aplicaciones nuevas en Azure, le recomendamos que planifique utilizar zonas de disponibilidad. Este enfoque ofrece flexibilidad para distribuir la aplicación y proporciona redundancia a los recursos de red que a menudo son fundamentales para que los clientes accedan a las VM subyacentes en última instancia. Para ver cómo funciona cada uno de estos enfoques, analicémoslos con más profundidad.

7.2 Redundancia de infraestructura con zonas de disponibilidad

Las zonas de disponibilidad son centros de datos físicamente separados que operan en instalaciones básicas independientes, como la conectividad de red y de energía. Cada región Azure compatible con zonas de disponibilidad proporciona tres zonas. Usted crea sus recursos en estas zonas y a través de ellas. La figura 7.2 muestra cómo se pueden distribuir los recursos Azure a través de zonas de disponibilidad.

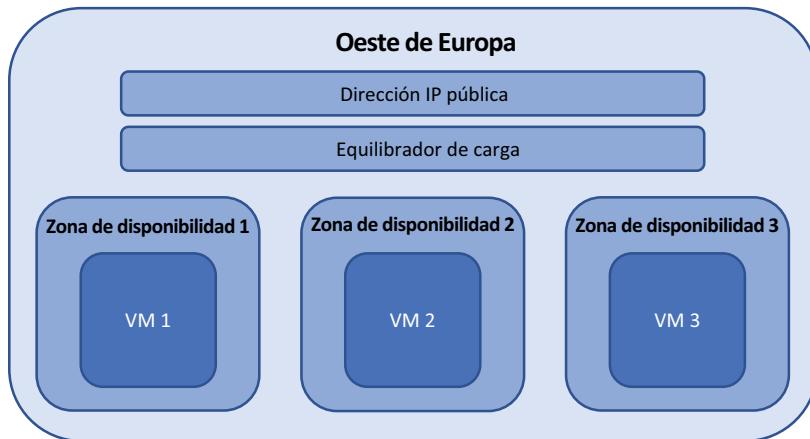


Figura 7.2 Una región Azure puede contener varias zonas de disponibilidad: centros de datos físicamente aislados que utilizan energía, red y enfriamiento independientes. Los recursos de red virtual Azure, como las direcciones IP públicas y los equilibradores de carga, pueden abarcar todas las zonas de una región para proporcionar redundancia no solo a las VM.

Con las zonas de disponibilidad, sus aplicaciones pueden tolerar que se caiga un centro de datos completo de Azure. Seguro tendría que suceder un acontecimiento importante para ocurriera esto, ¡pero incluso eso es posible!

En implementaciones de aplicaciones grandes, puede crear más de una VM en cada zona de disponibilidad. Varias máquinas virtuales en una zona de disponibilidad se distribuyen automáticamente en el hardware disponible dentro de la zona. No hay nada que tenga que configurar o que pueda controlar. Incluso si una actualización de mantenimiento o una falla del equipo dentro de una zona afectaran a todas las VM que se ejecutan en la zona, recuerde que las zonas están físicamente aisladas entre sí: las VM de otra zona continuarían funcionando.

Ahora, si tiene mucha mala suerte, ¿podrían sus VM en diferentes zonas tener actualizaciones de mantenimiento al mismo tiempo? Sí, pero es poco probable. Las zonas de una región tienen ciclos de actualización intercalados. Las actualizaciones se realizan en una zona; una vez completadas, se realizan en la siguiente zona. Las zonas de disponibilidad proporcionan un alto nivel de abstracción y redundancia, y debe mirar su aplicación en toda la implementación, no solo donde residen las VM en una zona.

La inclusión de los recursos de red virtuales en zonas de disponibilidad es mucho más importante de lo que puede parecer al principio. En la figura 7.3 se muestra lo que

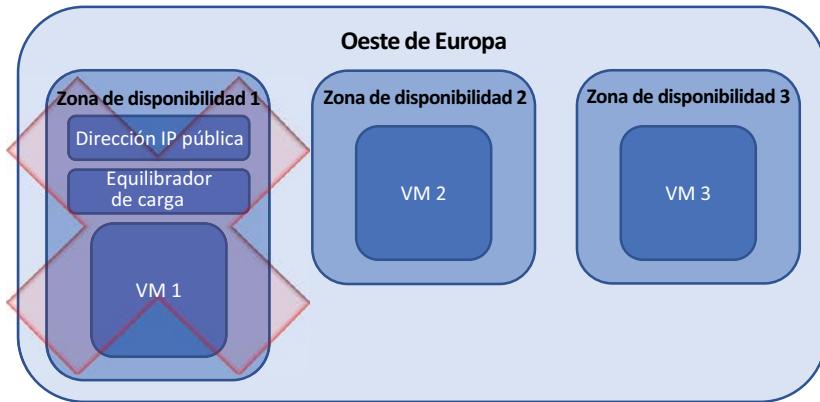


Figura 7.3 Cuando los recursos de red se conectan a un solo centro de datos o zona de Azure, una interrupción en esa instalación hará que el cliente no tenga acceso front-end a la aplicación entera. No importa que las otras VM continúen ejecutándose en otras zonas. Sin la conectividad de red para distribuir el tráfico de sus clientes, toda la aplicación queda no disponible.

sucedería si el centro de datos no estuviera disponible para los recursos de red, como una dirección IP pública y un equilibrador de carga que se ejecute a través de zonas de disponibilidad.

Hablaré más sobre equilibradores de carga en el capítulo 8, pero por ahora, todo lo que necesitas entender es que el equilibrador de carga distribuye el tráfico a través de todas las VM disponibles que se conectan a él. Las VM reportan su estado a intervalos establecidos y el equilibrador de carga deja de distribuir el tráfico a una VM que informa estar no disponible. Con un equilibrador de carga que funciona a través de zonas de disponibilidad, una interrupción en un centro de datos de Azure hará que las VM queden no disponibles y se saquen de la rotación del equilibrador de carga.

Una dirección IP pública que abarca zonas de disponibilidad proporciona un único punto de entrada para que los clientes lleguen a su equilibrador de carga y luego se distribuyan a una VM disponible. En una implementación de aplicación donde la dirección IP pública reside en un solo centro de datos de Azure, si ese centro de datos sufre un problema, ningún cliente puede acceder a la dirección IP pública. El cliente no puede utilizar su aplicación, aunque haya VM disponibles para atender las solicitudes de los clientes.

Los recursos que pueden utilizar las zonas de disponibilidad incluyen tanto los servicios zonales como los servicios con redundancia de zona:

- *Los servicios zonales* son para cosas como las máquinas virtuales, una dirección IP pública o un equilibrador de carga. Todo el recurso en sí funciona dentro de una zona determinada y puede funcionar por sí mismo si otra zona no está disponible.
- *Los servicios con redundancia de zona* son para recursos que pueden replicarse automáticamente entre zonas, como el almacenamiento con redundancia de zona y las bases de datos SQL. El recurso completo no se ejecuta en una zona determinada, sino que sus datos se distribuyen entre las zonas para que siga estando disponible si una zona tiene un problema.

La compatibilidad con la zona de disponibilidad está disponible para más de 20 servicios de Azure en más de diez regiones. El número de servicios y regiones que se integran con las zonas de disponibilidad sigue creciendo. Sin embargo, dadas las limitaciones de la región, puede haber ocasiones en las que la compatibilidad con la zona de disponibilidad no esté disponible para recursos básicos como las máquinas virtuales. En esos casos, hay otro tipo de redundancia de máquina virtual que se puede utilizar en cualquier región que vemos en la sección 7.2.1: Los conjuntos de disponibilidad.

7.2.1 Creación de recursos de red en una zona de disponibilidad

Para comenzar a ver algo de esta disponibilidad y redundancia en acción, creamos algunos recursos comunes, como una dirección IP pública y un equilibrador de carga, y luego las máquinas virtuales. El objetivo aquí es ver que no hay que hacer mucha configuración para aprovechar las zonas de disponibilidad en Azure. Estos son ejemplos sencillos, pero constituyen el núcleo de la mayoría de los entornos de aplicación que se implementan.

Las direcciones IP públicas y los equilibradores de carga se pueden crear en uno de los dos niveles disponibles: básico y estándar. La diferencia principal es que el nivel estándar permite que el recurso de red utilice zonas de disponibilidad. De forma predeterminada, una dirección IP pública estándar o equilibrador de carga es automáticamente redundante a nivel regional. No tiene que completar ninguna configuración adicional. La plataforma Azure almacena de forma centralizada los metadatos para el recurso dentro de la región que especifique y se asegura de que el recurso continúe funcionando si una zona no está disponible.

No se preocupe demasiado por lo que sucede con el equilibrador de carga y los recursos de red en este momento. Recuerde lo que dijimos al principio: estos dos o tres capítulos se construyen uno sobre el otro. En el capítulo 8, nos adentramos en los equilibradores de carga, y todo esto debería empezar a cobrar más sentido.

Pruébelo ahora

Complete los siguientes pasos para crear recursos de red redundantes en las zonas de disponibilidad:

- 1 Seleccione el icono de Cloud Shell en la parte superior del panel de Azure Portal.
- 2 Cree un grupo de recursos, como `azuremolchapter7az`:

```
az group create --name azuremolchapter7az --location westeurope
```

- 3 Cree una dirección IP pública estándar en el grupo de recursos. De forma predeterminada, se creará una dirección IP pública *básica* y se asignará a una sola zona. El parámetro `--sku standard` le indica a Azure que cree un recurso de zona transversal redundante:

```
az network public-ip create \
--resource-group azuremolchapter7az \
--name azpublicip \
--sku standard
```

- 4 Cree un equilibrador de carga que abarque las zonas de disponibilidad. Como ya lo mencionamos, de forma predeterminada se crearía un equilibrador de carga básico y se asignaría a una sola zona, pero no es el diseño de alta disponibilidad ideal para sus aplicaciones. Especifique un SKU de carga *estándar* para crear un equilibrador de carga con redundancia de zona, como se indica a continuación:

```
az network lb create \
--resource-group azremolchapter7az \
--name azloadbalancer \
--public-ip-address azpublicip \
--sku standard
```

7.2.2 Creación de VM en una zona de disponibilidad

Para crear una VM en una zona de disponibilidad, especifique la zona donde se ejecutará la VM. Para implementar varias VM, es ideal crear y utilizar una plantilla. La plantilla define y distribuye las zonas para cada una de las VM. A medida que crezca la demanda de los clientes por su pizzería en línea, puede actualizar la plantilla con el número de VM que quiere ahora y luego vuelva a implementar la plantilla. Las nuevas VM se distribuyen automáticamente a través de las zonas y no es necesario rastrear manualmente las zonas en las que se ejecutan las VM. En el laboratorio de fin del capítulo, se utiliza una plantilla para crear y distribuir automáticamente varias VM. Para ver el proceso lógico para especificar una zona para una VM, vamos a crear una VM y especificar manualmente la zona.

Pruébelo ahora

Complete los siguientes pasos para crear una VM en una zona de disponibilidad:

- 1 En Azure Portal, seleccione el ícono Cloud Shell en la parte superior del panel.
- 2 Cree una VM con el comando `az vm create` que usó en capítulos anteriores. Use el parámetro `--zone` para especificar si la VM se debe ejecutar en la zona 1, 2 o 3. El siguiente ejemplo crea una VM denominada `zonedvm` en la zona 3:

```
az vm create \
--resource-group azremolchapter7az \
--name zonedvm \
--image ubuntults \
--size Standard_B1ms \
--admin-username azremol \
--generate-ssh-keys \
--zone 3
```

Crear una VM demora unos minutos. Cuando termine el proceso, el resultado del comando indica la zona en la que se ejecuta la VM. También puede ver esta información con el comando `az vm show`:

```
az vm show \
--resource-group azremolchapter7az \
--name zonedvm \
--query zones
```

NOTA Los ejemplos en estos ejercicios "Pruébelo ahora" son simples, pero están diseñados para mostrarle que las zonas requieren poca configuración para poder utilizarse. No integró el equilibrador de carga con redundancia de zona con la VM, pero en el capítulo 8 compilará un entorno de aplicación más utilizable que se distribuye a través de zonas de disponibilidad. El objetivo aquí es mostrarle que la plataforma Azure maneja la redundancia y la distribución de sus recursos, para que pueda centrarse en la aplicación en sí.

7.3 Redundancia de VM con conjuntos de disponibilidad

Las zonas de disponibilidad son excelentes cuando se diseña la redundancia en un conjunto más amplio de recursos que conforman sus aplicaciones y cargas de trabajo. Le recomiendo que, en la medida de lo posible, las utilice para las nuevas cargas de trabajo. Sin embargo, hay ocasiones en las que no es necesario que toda la zona de recursos sea redundante. O puede que quiera crear máquinas virtuales en una región de Azure que no tenga actualmente soporte de zona de disponibilidad.

Si solo desea proporcionar redundancia para VM, los conjuntos de disponibilidad lo tienen cubierto. Su funcionamiento es comprobado, son confiables y están disponibles en todas las regiones. Los conjuntos de disponibilidad contienen un grupo lógico de VM que indica a la plataforma Azure el hardware subyacente en el que se ejecutan esas VM y que se debe seleccionar cuidadosamente. Si crea dos VM que se ejecutan en el mismo servidor físico y un servidor falla, ambas VM dejan de funcionar. Con potencialmente decenas de miles o más servidores físicos en un centro de datos de Azure, es muy improbable que tenga ambas VM en el mismo servidor, ¡pero es posible! Puede no ser un error, sino una actualización de mantenimiento que haga que el servidor físico esté no disponible brevemente.

¿Qué pasa si sus VM se ejecutan en el mismo rack, conectados al mismo equipo o red de almacenamiento? Vuelve al punto de falla único sobre el que hablamos al comienzo del capítulo.

Los conjuntos de disponibilidad permiten a la plataforma Azure crear sus VM en grupos lógicos denominados *dominios de error* y *dominios de actualización*. Estos dominios lógicos permiten a la plataforma Azure comprender los límites físicos de los grupos de hardware para asegurarse de que las VM se distribuyan uniformemente a través de ellas. Si parte del hardware sufre un problema, solo se verán afectadas unas pocas VM de su conjunto de disponibilidad. O si hay actualizaciones de mantenimiento que se apliquen al hardware físico, el mantenimiento afectará solo a algunas de las VM. En la figura 7.4, se muestra la relación entre el hardware físico y los dominios de error y los dominios de actualización lógicos dentro de un conjunto de disponibilidad.

Las zonas de disponibilidad hacen el mismo tipo de distribución bajo el capó, pero se abstraen y no se exponen. Incluso con los conjuntos de disponibilidad, no hay mucho que se pueda configurar. Pero es útil saber lo que ocurre en segundo plano.

7.3.1 Dominios de error

Un *dominio de error* es un grupo lógico de hardware en un centro de datos de Azure. Contiene hardware que comparte alimentación o equipo de red. Usted no controla lo que son estos dominios de error, y no hay nada que deba configurar en la VM. La plataforma Azure rastrea los dominios de error en los que se colocan sus VM y distribuye las nuevas

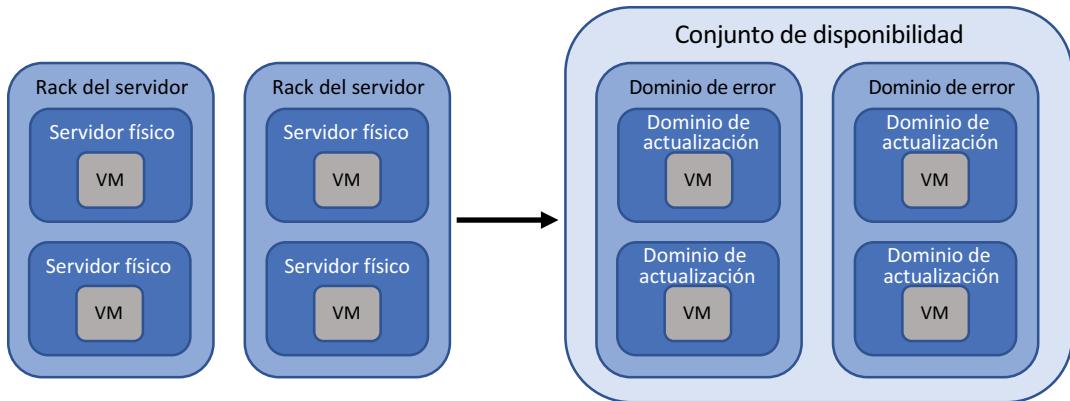


Figura 7.4 El hardware de un centro de datos de Azure se divide lógicamente en dominios de actualización y dominios de error. Estos dominios lógicos permiten a la plataforma Azure entender cómo distribuir sus VM a través del hardware subyacente para satisfacer sus requerimientos de redundancia. Este ejemplo es básico: un dominio de actualización probablemente contiene más de un servidor físico.

VM en estos dominios de error para que siempre disponga de VM si falla la alimentación o el conmutador de red.

Las VM que utilizan discos administrados (recuerde, ¡todas las VM deben utilizar discos administrados!) también respetan los límites lógicos de los dominios de error y distribución. La plataforma Azure asigna de forma lógica clústeres de almacenamiento a los dominios de error para garantizar que, a medida que las VM se distribuyan en grupos de hardware, los discos administrados también se distribuyan a través de hardware de almacenamiento. La redundancia de VM en el hardware del servidor no tendría sentido si existiera la posibilidad de que todos los discos administrados terminaran en un único clúster de almacenamiento. Y sí, los discos administrados también pueden utilizarse con las zonas de disponibilidad.

7.3.2 Dominios de actualización

Mientras que los dominios de error crean un grupo lógico de hardware para proteger contra fallas de hardware, los dominios de actualización protegen contra el mantenimiento rutinario. Para proporcionar esta protección, un dominio de error a su vez se divide lógicamente en dominios de actualización. Reitero, no hay nada que configurar aquí. Los dominios de actualización son una manera para que la plataforma Azure entienda cómo debe distribuir VM a través de su conjunto de disponibilidad.

Los ingenieros de Azure realizan mantenimiento (principalmente automatizado) y aplican actualizaciones en todo el hardware físico en un dominio de actualización y, a continuación, realizan el mismo mantenimiento en todo el hardware del siguiente dominio de actualización. Este trabajo de mantenimiento se intercala en los dominios de actualización para asegurarse de que las VM de un conjunto de disponibilidad no se ejecuten en hardware que al mismo tiempo esté en mantenimiento. Es el mismo tipo de proceso que vimos con las zonas de disponibilidad; la distribución de sus recursos significa que no puede tener un escenario en el que todo el hardware subyacente para sus recursos se esté actualizando al mismo tiempo.

No hay ninguna relación entre los dominios en varios conjuntos de disponibilidad. Los recursos físicos que componen los dominios de error y actualización en un conjunto de disponibilidad pueden no ser los mismos para un segundo conjunto de disponibilidad. Esto significa que si crea varios conjuntos de disponibilidad y distribuye sus VM a través de ellos, el dominio de error 1, por ejemplo, no siempre contiene el mismo hardware físico.

7.3.3 Distribución de las VM en un conjunto de disponibilidad

Vayamos paso a paso y veamos cómo se distribuyen las VM en los dominios lógicos de error y de actualización que componen un conjunto de disponibilidad. De esta manera, tiene varias VM que pueden ejecutar su pizzería, ¡y los clientes no pasarán hambre!

Pruébelo ahora

Para ver los conjuntos de disponibilidad en acción, complete los siguientes pasos para implementar una plantilla de Resource Manager:

- 1 Abra un navegador web y vaya a una plantilla de Resource Manager desde el repositorio de ejemplos de GitHub en <https://github.com/fouldsy/azure-mol-samples-2nd-ed/tree/master/> 07/availability-set y luego seleccione el botón Implementar en Azure. En este ejercicio, utilizará una plantilla para poder implementar rápidamente las VM y explorar cómo se distribuyen en el conjunto de disponibilidad.

Se abre Azure Portal y solicita algunos parámetros.

- 2 Seleccione para crear nuevo grupo de recursos y luego escriba un nombre, como azuremolchapter7. Seleccione una región y, a continuación, proporcione los datos de la clave SSH (puede obtenerla en este Cloud Shell con cat ~/ssh/id_rsa.pub).

La plantilla crea un conjunto de disponibilidad que contiene tres VM, las que se distribuyen a través de los dominios de error y actualización lógicos. Basándose en lo que aprendió sobre Resource Manager en el capítulo 6, esta plantilla utiliza la función copyIndex() para crear varias VM y NIC.

- 3 Para aceptar que desea crear los recursos detallados en la plantilla, marque la casilla "Acepto los términos y condiciones indicados anteriormente" y, a continuación, seleccione Comprar.

Toma unos minutos crear las tres VM en el conjunto de disponibilidad. Deje que continúe la implementación en el portal mientras lee el resto de la sección.

Cuando la plantilla comienza a implementarse, se crea un conjunto de disponibilidad y se asigna el número de dominios de actualización y de error que solicitó. Las siguientes propiedades se definieron en la plantilla de ejemplo:

```
"properties": {  
    "platformFaultDomainCount": "2",  
    "platformUpdateDomainCount": "5",  
    "managed": "true"  
}
```

Estas propiedades crean un conjunto de disponibilidad con dos dominios de error y cinco dominios de actualización, como se muestra en la figura 7.5, e indican que las VM deben utilizar discos administrados, así que recuerde respetar la distribución del disco. La región seleccionada para el conjunto de disponibilidad determina el número máximo de dominios de error y de actualización. Las regiones admiten 2 o 3 dominios de error y hasta 20 dominios de actualización.

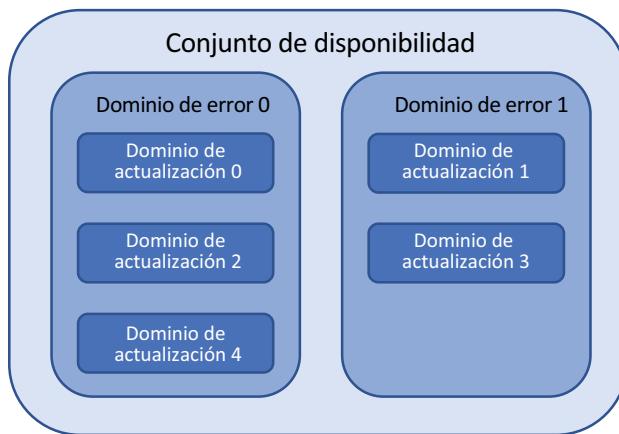


Figura 7.5 El conjunto de disponibilidad que implementa la plantilla de ejemplo contiene dos dominios de error y cinco dominios de actualización. El sistema de numeración se basa en cero. Los dominios de actualización se crean secuencialmente en los dominios de error.

A medida que crea más VM en un conjunto de disponibilidad, debe considerar la cantidad de dominios de actualización que se usarán. Por ejemplo, cinco dominios de actualización significa que hasta un 20 % de las VM puede no estar disponible a causa de mantenimiento:

- Digamos que tiene 10 VM en su conjunto de disponibilidad. Dos de esas VM pueden estar en mantenimiento al mismo tiempo. Si desea permitir que solo una VM esté en mantenimiento a la vez, necesitará crear 10 dominios de actualización. Cuantos más dominios de actualización crea, más tiempo estará la aplicación potencialmente en estado de mantenimiento.
- Continuemos con el ejemplo anterior de 10 máquinas virtuales en 10 dominios de actualización. Ahora existe la posibilidad de que se produzcan interrupciones en sus aplicaciones hasta que los 10 dominios de actualización hayan completado su ciclo de mantenimiento. Si solo tiene 5 dominios de actualización, ese plazo de mantenimiento se reduce. No es necesariamente malo tener un período de mantenimiento más largo; se trata más bien de cuál es su tolerancia para funcionar potencialmente a menos de la capacidad total.

Es importante recordar que estos dominios de actualización y ciclos de mantenimiento son los que realiza la propia plataforma Azure. También debe considerar sus propias necesidades de actualización y los plazos de mantenimiento.

Cuando se crea la primera VM, la plataforma Azure busca para ver dónde estaría disponible la primera posición de implementación. Este es el dominio de error 0 y dominio de actualización 0, como se muestra en la figura 7.6.

Cuando se crea la segunda VM, la plataforma Azure busca para ver dónde estaría disponible la siguiente posición de implementación. Ahora es el dominio de error 1 y dominio de actualización 1, como se muestra en la figura 7.7.

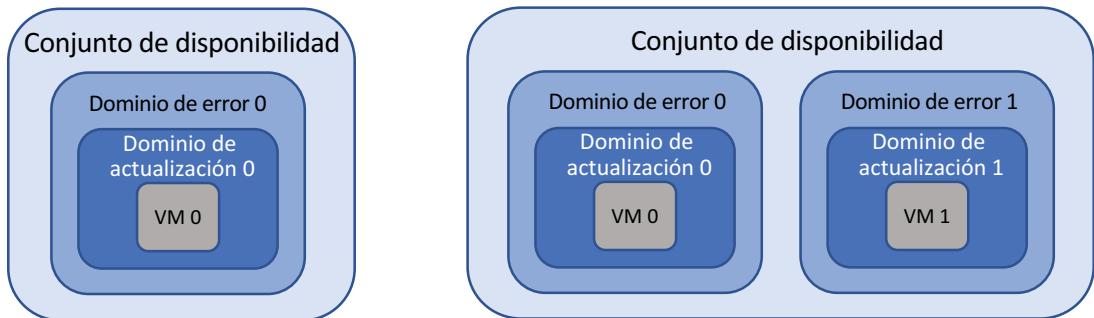


Figura 7.6 La primera VM se crea en el dominio de error 0 y dominio de actualización 0.

Figura 7.7 Habiendo creado una segunda VM, las VM se distribuyen uniformemente a través de los dominios de error y de actualización. Esto suele considerarse la cantidad mínima de redundancia necesaria para proteger sus aplicaciones.

Su plantilla crea tres VM, así que ¿qué cree que pasa después? La plataforma Azure vuelve a buscar dónde estaría disponible la siguiente posición de implementación. Ha creado solo dos dominios de error, así que la VM se crea de nuevo en el dominio de error 0, pero en un dominio de actualización diferente que la primera VM. La tercera VM se crea en el dominio de actualización 2, como se muestra en la figura 7.8.

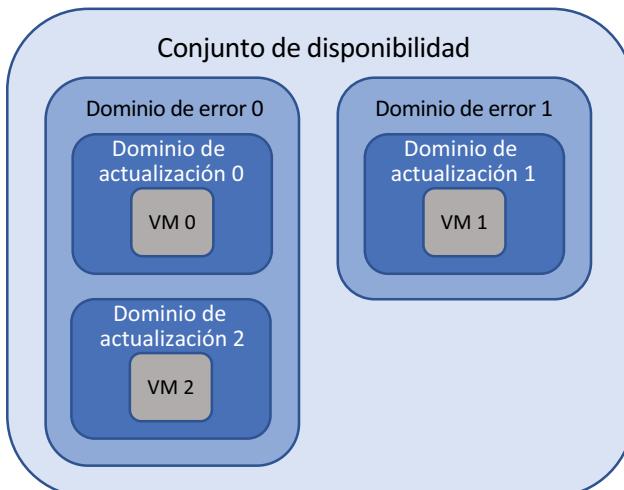


Figura 7.8 La tercera VM se crea de nuevo en el dominio de error 0, pero en el dominio de actualización 2. Aunque las VM 0 y 2 comparten el riesgo de falla de hardware, están en diferentes dominios de actualización y, por lo tanto, no estarán en mantenimiento normal al mismo tiempo.

Las VM 0 y 2 están en el mismo dominio de error, por lo que una falla de hardware podría posiblemente afectar a ambas VM. Pero el mantenimiento rutinario solo afecta a una de esas VM a la vez, porque están distribuidas a través de dominios de actualización. Si continúa y crea más VM, la plataforma Azure continuará distribuyéndolas a través de diferentes dominios de error y de actualización. Cuando se utilizan los cinco dominios de actualización, la sexta VM se crea de nuevo en el dominio 0 de actualización y el ciclo continúa.

7.3.4 Visualización de distribución de las VM en un conjunto de disponibilidad

Ahora que entiende la teoría sobre cómo se distribuyen las VM en los dominios de error y de actualización en un conjunto de disponibilidad, veamos qué ha sucedido con la implementación de su plantilla de Resource Manager.

Pruébelo ahora

Complete los siguientes pasos para ver cómo se distribuyen las VM en un conjunto de disponibilidad:

- 1 Busque y seleccione Grupos de recursos en la barra de navegación del lado izquierdo de Azure Portal.
- 2 Elija el grupo de recursos que creó para la implementación de la plantilla, como azuremolchapter7.
- 3 Seleccione el conjunto de disponibilidad de la lista de recursos, como azuremolavail-abilityset.

En la ventana Información general, hay una lista de VM y sus dominios asociados a errores y actualizaciones, como se muestra en la figura 7.9.

NAME	STATUS	FAULT DOMAIN	UPDATE DOMAIN
vm0	Running	1	1
vm1	Running	0	2
vm2	Running	0	0

Figura 7.9 El conjunto de disponibilidad enumera las VM que contiene y muestra el dominio de error y el dominio de actualización para cada VM. Esta tabla le permite visualizar cómo se distribuyen las VM en los dominios lógicos.

Si es observador, puede ver que las VM no se alinean perfectamente con el orden esperado de los dominios de error y actualización. ¿Hay algún error? Probablemente no. Si examina el ejemplo en la figura 7.9 y lo compara con los conceptos que aprendió, esperaría que las VM se distribuyeran como se muestra en la tabla 7.1.

Tabla 7.1 VM de un conjunto de disponibilidad creadas y distribuidas de manera secuencial entre dominios

Nombre	Dominio de error	Dominio de actualización
vm0	0	0
vm1	1	1
vm2	0	2

Entonces, ¿qué salió mal? Nada. Vuelva a pensar en cómo Resource Manager crea recursos a partir de una plantilla. La plataforma Azure no espera a que se cree la primera VM antes de que se pueda crear la segunda. Las tres VM se crean al mismo tiempo. Entonces, puede haber fracciones de segundo de diferencia en que la VM se asocia primero a un conjunto de disponibilidad. No importa cuál sea el orden, porque no puede controlar lo que representan los dominios de error y actualización subyacentes. Depende de la plataforma Azure. Solo tiene que asegurarse de que sus VM *estén* distribuidas, no *dónde*.

No, prefiero algo más organizado

Si el comportamiento de creación en serie de VM le molesta y *debe* distribuir las VM en un orden organizado, puede indicarle a Resource Manager que cree VM en *serie* en lugar de hacerlo en *paralelo*. En este modo, las VM se crean una tras otra, por lo que aumenta el tiempo de implementación. Para habilitar este comportamiento en serie, use "mode": "serial" en su plantilla como parte de la función copyIndex(). Eso debiera distribuir las VM de una manera organizada y secuencial.

7.4 Laboratorio: Implementación de VM altamente disponibles desde una plantilla

Este laboratorio combina y refuerza lo que aprendió en el capítulo 6 sobre Azure Resource Manager y plantillas con zonas de disponibilidad. Tómese un tiempo para examinar la plantilla de inicio rápido de ejemplo en este ejercicio y ver cómo puede utilizar la lógica y las funciones para distribuir varias VM entre zonas. No basta con implementar la plantilla y continuar, mire cómo la plantilla se basa en las funciones introducidas en el capítulo 6.

¿Qué es una cuota?

En Azure, las cuotas predeterminadas de su suscripción le impiden implementar accidentalmente un montón de recursos y olvidarse de ellos, lo que le costaría mucho dinero. Por lo general, estas cuotas varían según el tipo de recurso y el tipo de suscripción y se aplican a nivel regional. Puede ver una lista completa de cuotas en <http://mng.bz/ddcx>.

Cuando comience a crear varias VM en los próximos capítulos, podría tener problemas de cuotas. También puede encontrarse con problemas de cuotas si no ha eliminado recursos de los capítulos y ejercicios anteriores. Las cuotas son un buen sistema que lo mantiene al tanto de su uso de recursos. Es posible que los mensajes de error no sean claros, pero si ve texto de error en las líneas de esa es suficiente indicación de que necesita solicitar un aumento de sus cuotas.

```
Operation results in exceeding quota limits of Core.  
Maximum allowed: 4, Current in use: 4, Additional requested: 2.
```

solicitar un aumento de sus cuotas. No hay nada complicado y no es algo único de Azure. Puede ver su cuota actual para una región determinada de la siguiente manera:

```
az vm list-usage --location eastus
```

Si tiene problemas con este laboratorio, elimine los dos primeros grupos de recursos que creó en este capítulo, como `azuremolchapter7` y `azuremolchapter7az`. Si tiene un conjunto de cuotas predeterminado bajo, las cuatro VM que se encuentran en esos grupos de recursos pueden impedir que complete satisfactoriamente este ejercicio.

Para solicitar un aumento de sus cuotas para una región, siga los pasos descritos en <http://mng.bz/Xq2f>.

Revisemos e implementemos una plantilla de ejemplo que incluye varias VM en zonas de disponibilidad.

- 1 En un navegador web, abra el archivo JSON en <https://github.com/Azure/azure-quick-start-templates/blob/master/201-multi-vm-lb-zones/azuredeploy.json> y busque el siguiente texto:

```
Microsoft.Compute/virtualMachines
```

La sección de VM se ve similar a la que usó en el capítulo 6, pero observe el valor de la propiedad para zonas. Esta sección combina algunas funciones disponibles en las plantillas para seleccionar las zonas 1, 2 o 3 a medida que se crea la VM. De esta forma, no es necesario rastrear manualmente la VM que se ejecuta en qué zona y cómo se implementan las VM adicionales.

- 2 En su navegador web, busque cada una de las siguientes opciones para ver las secciones de la dirección IP pública y el equilibrador de carga:

```
Microsoft.Network/publicIPAddresses  
Microsoft.Network/loadBalancers
```

Ambos recursos utilizan SKU estándar, que proporciona redundancia de zona predeterminadamente. No hay ninguna configuración adicional que hacer para que funcione. Veámoslo en acción.

- 3 En el navegador web, abra la plantilla de inicio rápido en <http://mng.bz/O69a> y seleccione el botón Implementar en Azure.
- 4 Cree o seleccione un grupo de recursos y luego escriba un nombre de usuario y una contraseña para las VM.
- 5 Escriba un nombre DNS único, como `azuremol`.
- 6 Seleccione si crear una VM Linux o Windows. Las VM Windows tardan un poco más en crearse.

- 7 Especifique cuántas VM crear, por ejemplo, 3.
- 8 Marque la casilla para aceptar los términos y condiciones de implementación de la plantilla y seleccione Comprar, como se muestra en la figura 7.10.

VMs in Availability Zones with a Load Balancer and NAT
Azure quickstart template

TEMPLATE

201-multi-vm-lb-zones
8 resources

Edit template Edit parameters Learn more

BASICS

- * Subscription Azure
- * Resource group Create new Use existing azuremolchapter7lab
- * Location Central US

SETTINGS

- Location CentralUS
- * Admin Username azuremol
- * Admin Password *****
- * Dns Name azuremol
- Windows Or Ubuntu Ubuntu
- Number Of Vms 3

TERMS AND CONDITIONS

Template information | Azure Marketplace Terms | Azure Marketplace

By clicking "Purchase," I (a) agree to the applicable legal terms associated with the offering; (b) authorize Microsoft to charge or bill my current payment method for the fees associated with the offering(s), including applicable taxes, with the same billing frequency as my Azure subscription, until I discontinue use of the offering(s); and (c) agree that, if the deployment involves 3rd party offerings, Microsoft may share my contact information and other details of such deployment with the publisher of that offering.

Pin to dashboard

Purchase

Figura 7.10 Para implementar la plantilla de zona de disponibilidad en Azure Portal, especifique un grupo de recursos, un nombre de usuario y una contraseña y, a continuación, el tipo de SO y la cantidad de VM que desee crear. La plantilla usa bucles, `copyIndex()`, `dependsOn`, variables y parámetros, como se vio en el capítulo 6.

Cuando se hayan creado las VM, utilice Azure Portal o el comando `az vm show` para ver cómo se distribuyeron las VM entre zonas. Si tiene curiosidad sobre lo que hace el resto de la plantilla con los recursos de red, el capítulo 8 profundiza en los equilibradores de carga.

Limpieza en el pasillo 3

Recuerde que al comienzo del libro le dije que se asegurara de ir limpiando recursos para minimizar el costo de los créditos de Azure gratuitos. Le aconsejamos encarecidamente que elimine los grupos de recursos creados en este capítulo. En el siguiente par de capítulos continuará creando varias instancias de VM y aplicaciones web, así que asegúrese de mantener los costos y las cuotas bajo control.

Cada vez que inicie sesión en Azure Portal, debiera recibir una notificación emergente que le permitirá conocer el estado de sus créditos de Azure. Si ve que el monto de su crédito se reduce en gran cantidad día a día, examine qué grupos de recursos podría haber olvidado eliminar.

Aplicaciones de equilibrio de carga

Un componente importante de las aplicaciones altamente disponibles es cómo distribuir el tráfico en todas las VM. En el capítulo 7, aprendió la diferencia entre los conjuntos de disponibilidad y las zonas de disponibilidad, y cómo puede crear varias VM en los centros de datos o regiones de Azure para proporcionar redundancia a la aplicación. Aunque tenga todas estas VM altamente disponibles y distribuidas, eso no ayuda si solo una VM recibe todo el tráfico de los clientes.

Los equilibradores de carga son recursos de red que reciben el tráfico de aplicaciones entrante de sus clientes, examinan el tráfico para aplicar filtros y reglas de equilibrio de carga y luego distribuyen las solicitudes a través de un grupo de VM que ejecutan la aplicación. En Azure, hay un par de maneras de equilibrar la carga del tráfico, como si necesita descargar SSL en aplicaciones grandes que utilizan tráfico de red cifrado. En este capítulo, aprenderá acerca de los diversos componentes del equilibrador de carga y cómo configurar las reglas de tráfico y los filtros, además de distribuir el tráfico a las VM. Se basa en los componentes de alta disponibilidad del capítulo 8 y se prepara para el capítulo 9 sobre cómo escalar los recursos.

8.1 Componentes del equilibrador de carga de Azure

Los equilibradores de carga de Azure pueden trabajar en dos niveles: capa 4, donde solo se examina y distribuye el tráfico de red (realmente, la capa de transporte) y la capa 7, donde se reconocen los datos de la aplicación dentro del tráfico de red para ayudar a determinar la distribución de los datos. Ambos niveles del equilibrador de carga funcionan de la misma manera, como se muestra en la figura 8.1.

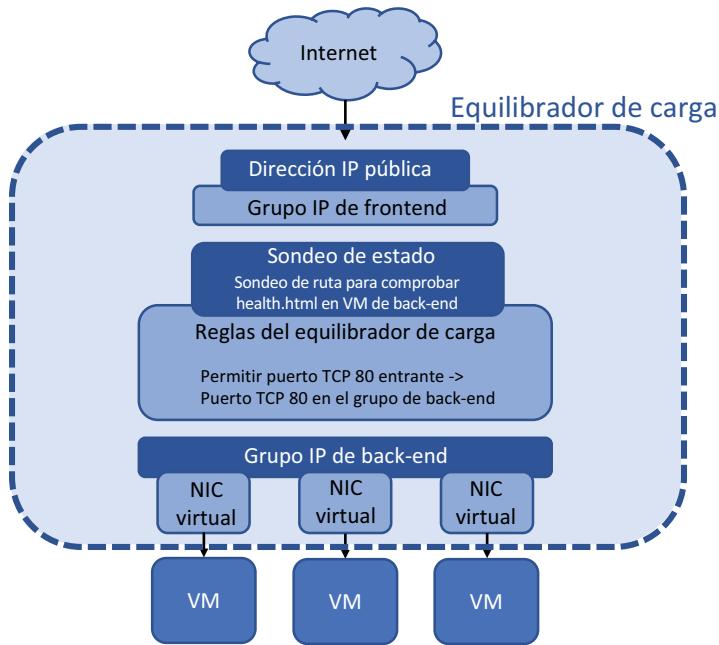


Figura 8.1 El tráfico desde Internet ingresa al equilibrador de carga a través de una dirección IP pública conectada a un grupo IP de front-end. El tráfico se procesa mediante reglas del equilibrador de carga que determinan cómo y dónde debe reenviar el tráfico. Los sondeos de estado conectados a las reglas garantizan que el tráfico solo se distribuya a nodos en buen estado. Luego, un grupo de back-end de NIC virtuales conectados a las VM recibe el tráfico distribuido por las reglas del equilibrador de carga.

Un equilibrador de carga consta de unos pocos componentes principales:

- *Grupo IP de front-end*: punto de entrada al equilibrador de carga. Para permitir el acceso desde Internet, se puede conectar una dirección IP pública al grupo IP de front-end. Se pueden conectar direcciones IP privadas para los equilibriadores de carga internos.
- *Sondeos de estado*: controlan el estado de las VM conectadas. Para asegurarse de que el tráfico solo se distribuya a VM en buen estado y con capacidad de respuesta, se realizan controles periódicamente para confirmar que la VM responda correctamente al tráfico.
- *Reglas del equilibrador de carga*: distribuyen el tráfico a las VM. Cada paquete entrante se compara con las reglas que definen los protocolos y puertos entrantes y, luego, se distribuyen a través de un conjunto de VM asociadas. Si no hay reglas que coincidan con el tráfico entrante, el tráfico se elimina.
- *Reglas de traducción de direcciones de red (NAT)*: pueden enrutar el tráfico directamente a VM específicas. Por ejemplo, si desea proporcionar acceso remoto a través de SSH o RDP, puede definir reglas NAT para reenviar el tráfico desde un puerto externo a una sola VM.
- *Grupo IP de back-end*: lugar donde se conectan las VM que ejecutan la aplicación. Las reglas del equilibrador de carga están asociadas con los grupos de back-end. Puede crear diferentes grupos de back-end para diferentes partes de sus aplicaciones.

Azure Application Gateway: equilibrador de carga avanzado

Los equilibradores de carga de Azure pueden trabajar en la capa de red o en la capa de aplicación. Este capítulo se centra en el equilibrador de carga normal de Azure, que funciona en la capa de red (capa 4, o protocolo de transporte). En esta capa, el tráfico se examina y distribuye, pero el equilibrador de carga no tiene ningún contexto de lo que significa el tráfico o las aplicaciones que se ejecutan.

Application Gateway de Azure: es un equilibrador de carga que funciona en la capa de aplicación (capa 7). Application Gateway obtiene información de la aplicación que se ejecuta en la VM y puede administrar los flujos de tráfico de formas más avanzadas. Una de las principales ventajas de Application Gateway es la capacidad de manejar tráfico web encriptado y HTTPS.

Cuando se equilibra la carga de los sitios web con certificados SSL, se puede descargar el proceso que verifica y descifra el tráfico de los servidores web. En sitios web con mucho tráfico SSL, el proceso para verificar y descifrar el tráfico puede consumir gran parte del tiempo de proceso en las VM o aplicaciones web. Application Gateway puede verificar y descifrar el tráfico, pasar la solicitud web pura a los servidores web y luego volver a cifrar el tráfico recibido de los servidores web y devolverlo al cliente.

Application Gateway ofrece algunas otras funciones del equilibrador de carga más avanzadas, como la capacidad de distribuir tráfico a través de cualquier punto de conexión IP en lugar de una única VM de Azure. A medida que compila aplicaciones que utilizan más que VM, estas reglas de distribución avanzadas pueden serle de utilidad. Los mismos conceptos básicos se aplican que con un equilibrador de carga normal; en este capítulo nos centraremos en esto para que entienda cómo funciona todo de manera conjunta en Azure.

8.1.1 Creación de grupos IP de front-end

En capítulos anteriores, creó VM que tenían una dirección IP pública asignada directamente a ellas. Utilizó esta dirección IP pública para luego acceder a la VM con una conexión remota como SSH o RDP, o utilizó un navegador web para acceder a un sitio web que se ejecutaba en la VM. Cuando se utiliza un equilibrador de carga, ya no se conecta directamente a las VM. En lugar de ellos, para permitir que el tráfico llegue a su equilibrador de carga y se distribuya a las VM, se debe asignar una o más direcciones IP a la interfaz externa de un equilibrador de carga.

Los equilibradores de carga pueden funcionar en uno de dos modos:

- *Equilibrador de carga de Internet:* tiene una o más direcciones IP *públicas* conectadas al grupo IP de front-end. Un equilibrador de carga de Internet recibe directamente el tráfico de Internet y lo distribuye a las VM de back-end. Un ejemplo común son los servidores web front-end a los que los clientes acceden directamente a través de Internet.
- *Equilibrador de carga interno:* tiene una o más direcciones IP *privadas* conectadas al grupo IP de front-end. Un equilibrador de carga interno funciona dentro de una red virtual Azure, como para las VM de bases de datos back-end. Normalmente, no expone bases de datos de back-end o niveles de aplicación al mundo exterior. En lugar de ello, un conjunto de servidores web front-end se conecta a un equilibrador de carga interno que distribuye el tráfico sin ningún tipo de acceso público directo. La figura 8.2 muestra cómo un equilibrador de carga interno puede distribuir el tráfico a las VM de back-end que están detrás de un equilibrador de carga orientado al público y VM web de front-end.

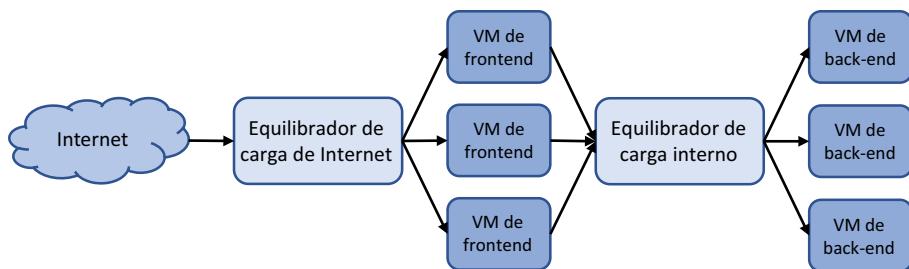


Figura 8.2 Se puede utilizar un equilibrador de carga de Internet para distribuir el tráfico a las VM de front-end que ejecutan su sitio web, que luego se conectan a un equilibrador de carga interno para distribuir el tráfico en el nivel de base de datos de las VM. El equilibrador de carga interno no es accesible al público y solo se puede acceder a él desde las VM de front-end dentro de la red virtual Azure.

El modo para el equilibrador de carga no cambia el comportamiento del grupo IP de front-end. Se asignan una o más direcciones IP que se utilizan cuando se solicita el acceso al equilibrador de carga. Se pueden configurar tanto las direcciones IPv4 como IPv6 para el grupo IP de front-end, lo que le permite configurar las comunicaciones IPv6 de un extremo a otro entre los clientes y las VM a medida que el tráfico fluye dentro y fuera del equilibrador de carga.

Pruébelo ahora

Para entender cómo funcionan los componentes del equilibrador de carga, siga los siguientes pasos para crear un equilibrador de carga y un grupo IP de front-end:

- 1 Abra Azure Portal y seleccione el icono Cloud Shell en la parte superior del panel.
- 2 Cree un grupo de recursos con `az group create`.

Especifique un nombre de grupo de recursos, como `azuremolchapter8`, y una ubicación:

```
az group create --name azuremolchapter8 --location westeurope
```

A medida que continúe compilando sobre el capítulo 7 y utilice zonas de disponibilidad, tenga cuidado con la región que seleccione para asegurarse de que la zona de disponibilidad sea compatible.

- 3 Cree una dirección IP pública con `az network public-ip create`.

En el capítulo 7, aprendió que las zonas de disponibilidad proporcionan redundancia a los recursos de red, así que cree una dirección IP pública estándar con la VM y especifique un nombre, como `publicip`:

```
az network public-ip create \
--resource-group azuremolchapter8 \
--name publicip \
--sku standard
```

Para crear una dirección IP pública IPv6, puede agregar `--version IPv6` al comando anterior. Para estos ejercicios, puede usar direcciones IPv4.

- 4 Cree el equilibrador de carga y asigne la dirección IP pública al grupo IP de front-end. Para agregar la dirección IP pública, especifique el parámetro `--public-ip-address`. Si quería crear un equilibrador de carga interno, tendría que haber utilizado el parámetro `--private-ip-address`.

Al igual que con la dirección IP pública, cree un equilibrador de carga estándar con redundancia de zona que funcione a través de zonas de disponibilidad:

```
az network lb create \
    --resource-group azurermolchapter8 \
    --name loadbalancer \
    --public-ip-address publicip \
    --frontend-ip-name frontendpool \
    --backend-pool-name backendpool \
    --sku standard
```

En las próximas páginas profundizaremos en los grupos de back-end.

8.1.2 Creación y configuración de sondeos de estado

Si una de las VM que ejecuta su aplicación tiene un problema, ¿cree que el equilibrador de carga debiera continuar distribuyendo el tráfico a esa VM? Un cliente que intenta acceder a su pizzería puede ser dirigido a esa VM y no podrá hacer su pedido. Un equilibrador de carga supervisa el estado de las VM y puede eliminar aquellas que tienen problemas. El equilibrador de carga continúa monitoreando el estado y vuelve a agregar la VM al grupo para distribución del tráfico una vez que la VM vuelve a responder correctamente.

Un sondeo de estado puede funcionar de dos modos:

- *Basado en puerto*: el equilibrador de carga comprueba la respuesta de la VM en un puerto y protocolo específicos, como puerto TCP 80. Mientras la VM responda a los sondeos de estado en el puerto TCP 80, la VM permanecerá en la distribución de tráfico del equilibrador de carga. De lo contrario, la VM se elimina de la distribución de tráfico del equilibrador de carga, como se muestra en la figura 8.3. Este modo no garantiza que la VM sirva el tráfico como se esperaba, solo comprueba que la conectividad de red y el servicio de destino devuelvan una respuesta.
- *Basado en ruta de HTTP*: una página personalizada, como `health.html`, se escribe y se coloca en cada VM. Esta comprobación de estado personalizada se puede utilizar para verificar el acceso a un almacén de imágenes o a una conexión de base de datos. En este modo, la VM solo permanece en la distribución de tráfico del equilibrador de carga cuando la página de comprobación de estado devuelve una respuesta código HTTP 200, como se muestra en la figura 8.4. Con un sondeo de estado basado en puerto, el servidor web real puede ejecutarse pero no tener conexión a la base de datos. Con una página de comprobación de estado personalizada, el equilibrador de carga puede confirmar que la VM es capaz de servir el tráfico real a los clientes.

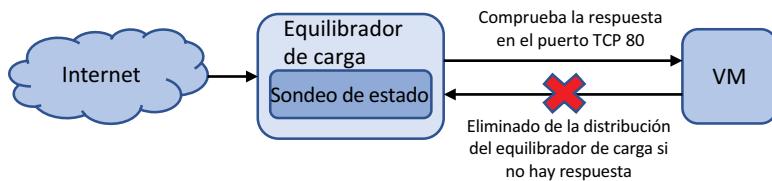


Figura 8.3 Un sondeo de estado de un equilibrador de carga basada en puerto comprueba la respuesta de una VM en un puerto y protocolo definidos. Si la VM no responde dentro del umbral dado, esta se elimina de la distribución de tráfico del equilibrador de carga, y cuando comienza a responder correctamente de nuevo, el sondeo de estado detecta el cambio y vuelve a agregar la VM a la distribución del tráfico del equilibrador de carga.

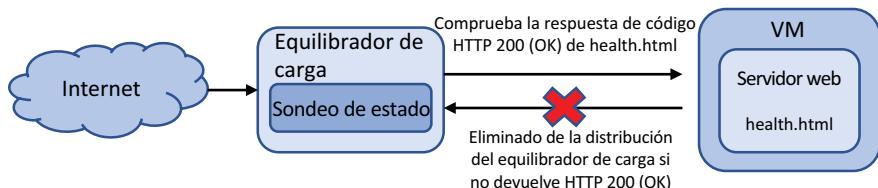


Figura 8.4 Una VM que ejecuta un servidor web y tiene una página personalizada `health.html` permanecerá en la distribución de tráfico del equilibrador de carga, siempre que el sondeo de estado reciba una respuesta de código HTTP 200 (OK). Si el proceso de servidor web encuentra un problema y no puede mostrar las páginas solicitadas, esas páginas se eliminan de la distribución de tráfico del equilibrador de carga. Este proceso proporciona una comprobación más exhaustiva del estado del servidor web que los sondeos de estado basados en puerto.

Se requiere trabajo adicional para crear la página de comprobación de estado personalizada, pero la experiencia mejorada para el cliente hace que valga la pena. La página de comprobación de estado no tiene que ser complicada. Podría ser una página HTML básica que se utilice para confirmar que el propio servidor web puede atender páginas. Sin la página de comprobación de estado, si el proceso del servidor web tiene un problema, la VM aún estaría disponible en el puerto TCP 80, y el sondeo de estado basado en puerto creería que la VM está en buen estado. Un sondeo de estado basado en ruta HTTP requiere que el servidor web devuelva correctamente una respuesta HTTP. Si el proceso de servidor web se cae o falla, no se envía una respuesta HTTP, por lo que la VM se elimina de la distribución de tráfico del equilibrador de carga.

La frecuencia con la que el sondeo de estado comprueba la VM y cuál es la respuesta, también se puede configurar mediante dos parámetros:

- *Intervalo*: permite definir la frecuencia con la que el sondeo de estado comprueba el estado de la VM. De forma predeterminada, el sondeo de estado comprueba el estado cada 15 segundos.
- *Umbral*: permite definir la cantidad de errores de respuesta consecutivas que puede recibir el sondeo de estado antes de que se elimine la VM de la distribución de tráfico del equilibrador de carga. De forma predeterminada, el sondeo de estado tolera dos fallas consecutivas antes de que la VM se elimine de la distribución de tráfico del equilibrador de carga.

Pruébelo ahora

Complete los siguientes pasos para crear un sondeo de estado para su equilibrador de carga, como en la figura 8.4.

- 1 Abra Azure Portal y seleccione el icono Cloud Shell en la parte superior del panel.
- 2 Especifique un nombre para el sondeo de estado, como `healthprobe`. Para configurar el sondeo de estado para un servidor web, especifique el puerto 80 de HTTP y, a continuación, defina una página de comprobación de estado personalizada en `health.html`. En la sección 8.2, creará esta página de comprobación de estado en sus máquinas virtuales. Para mostrar cómo se puede configurar el intervalo y el umbral de la respuesta de sondeo de estado, defina un intervalo de 10 segundos y un umbral de tres fallas consecutivas:

```
az network lb probe create \
--resource-group azurermchapter8 \
--lb-name loadbalancer \
--name healthprobe \
--protocol http \
--port 80 \
--path health.html \
--interval 10 \
--threshold 3
```

Después de crear el sondeo de estado, ¿cómo se hace para que compruebe el estado de las VM? Los sondeos de estado están asociados a reglas del equilibrador de carga. Se puede utilizar el mismo sondeo de estado con varias reglas del equilibrador de carga. ¿Recuerda el capítulo 5, cuando creó reglas y grupos de seguridad de red (NSG)? Esos NSG se pueden asociar a varias VM o subredes virtuales de red. Una relación similar "una a muchas" se aplica a los sondeos de estado.

Veamos cómo poner su sondeo de estado a trabajar y crear reglas del equilibrador de carga.

8.1.3 Definición de la distribución de tráfico con reglas del equilibrador de carga

Cuando el tráfico se dirige a través del equilibrador de carga a las VM de back-end, puede definir qué condiciones hacen que el usuario se dirija a la misma VM. Es posible que desee que el usuario conserve una conexión con la misma VM durante una sola sesión, o que permita que devuelvan y mantengan su afinidad de VM basándose en la dirección IP de origen. La figura 8.5 muestra un ejemplo del modo de afinidad de sesión predeterminado.

En el modo de afinidad de sesión, el flujo de tráfico se controla mediante un hash de 5 tuplas que utiliza la dirección IP de origen, el puerto de origen, la dirección IP de destino, el puerto de destino y el tipo de protocolo. Básicamente, para cada solicitud que un usuario hace a su servidor web en el puerto TCP 80, este se dirige a la misma VM de back-end por la duración de esa sesión.

¿Qué sucede si el cliente cierra la sesión del explorador? La próxima vez que se conecte, se iniciará una nueva sesión. Dado que el equilibrador de carga distribuye el tráfico en todas las VM en buen estado del grupo IP de back-end, es posible que el usuario se conectaría a la misma VM; pero cuantas más VM tenga el grupo IP de back-end, mayor será la posibilidad de que se conecte a una VM diferente.

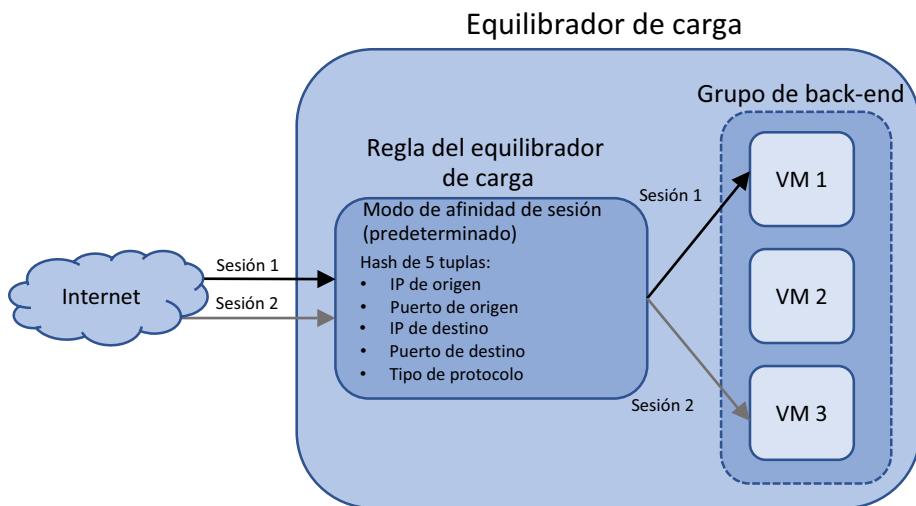


Figura 8.5 Con el modo de afinidad de sesión, el usuario se conecta a la misma VM de back-end solo por la duración de su sesión.

Como dueño y desarrollador de la aplicación, puede que quiera que el usuario se conecte a la misma VM que se conectó antes al iniciar otra sesión. Por ejemplo, si la aplicación controla las transferencias de archivos o utiliza UDP en lugar de TCP, es probable que quiera que la misma VM continúe procesando las solicitudes de los usuarios. En estos casos, puede configurar las reglas del equilibrador de carga para afinidad de IP de origen. La figura 8.6 muestra un ejemplo del modo de afinidad de IP de origen.

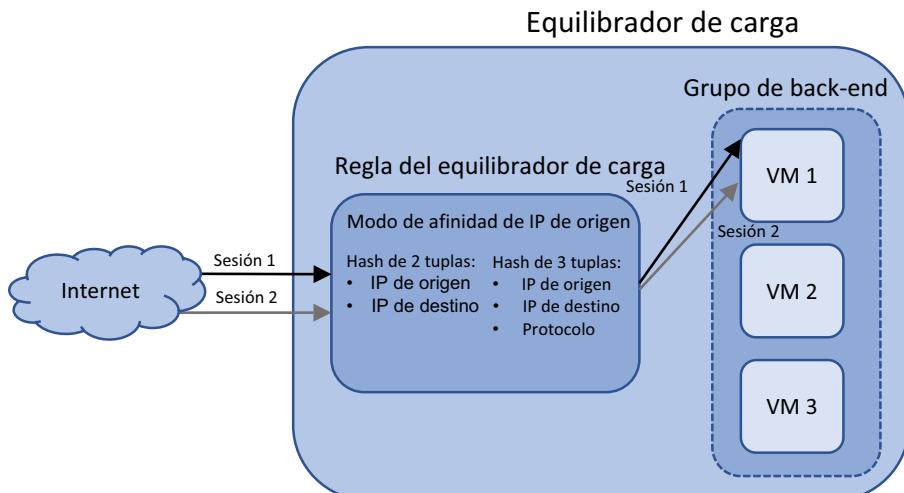


Figura 8.6 Al configurar las reglas del equilibrador de carga para utilizar el modo de afinidad de IP de origen, el usuario puede cerrar y luego iniciar una nueva sesión, pero continuar conectándose a la misma VM de back-end. El modo de afinidad de IP de origen puede utilizar un hash de 2 tuplas que utilice la dirección IP de origen y destino, o un hash de 3 tuplas que también utilice el protocolo.

Pruébelo ahora

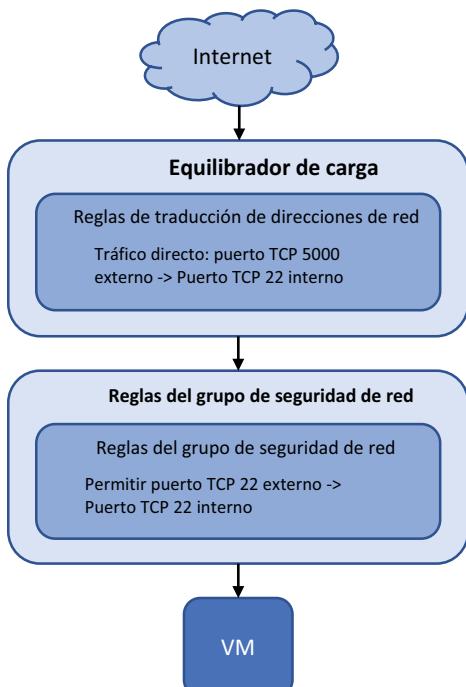
Complete los siguientes pasos para crear una regla del equilibrador de carga que utilice un sondeo de estado:

- 1 Abra Azure Portal y seleccione el icono Cloud Shell en la parte superior del panel.
- 2 Para crear una regla del equilibrador de carga, especifique un nombre para la regla, como httprule.
- 3 Proporcione el puerto externo en el que se recibe el tráfico y el puerto interno al cual distribuir el tráfico. En este ejemplo básico, el tráfico se recibe en el puerto 80 y luego se distribuye al puerto 80:

```
az network lb rule create \
--resource-group azuremolchapter8 \
--lb-name loadbalancer \
--name httprule \
--protocol tcp \
--frontend-port 80 \
--backend-port 80 \
--frontend-ip-name frontendpool \
--backend-pool-name backendpool \
--probe-name healthprobe
```

Si ejecuta varios sitios web en una VM que responde en diferentes puertos, una regla determinada podría dirigir el tráfico a un sitio web específico en la VM.

8.1.4 Enrutamiento de tráfico directo con reglas de traducción de direcciones de red



Las reglas del equilibrador de carga distribuyen el tráfico a través de los grupos de back-end de las VM, por lo que no hay garantía de que se pueda conectar a una VM determinada para fines de mantenimiento o administración. ¿Cómo se puede conectar a una VM específica que está detrás de un equilibrador de carga? Una última parte de la configuración del equilibrador de carga que hay que tener en cuenta son las reglas de traducción de direcciones de red (NAT), que permiten controlar el flujo de tráfico específico para dirigirlo a una única máquina virtual. En la figura 8.7 se muestra cómo las reglas NAT reenvían tráfico específico a VM individuales.

Figura 8.7 El tráfico en el equilibrador de carga es procesado por reglas NAT. Si un protocolo y un puerto coinciden con una regla, el tráfico se reenvía a la VM de back-end definida. No tiene sondeos de estado asociados, por lo que el equilibrador de carga no comprueba si la VM es capaz de responder antes de reenviar el tráfico. El tráfico sale del equilibrador de carga y luego es procesado por reglas NSG. Si el tráfico está permitido, se pasa a la VM.

Las reglas NAT funcionan junto con las reglas NSG. La VM solo puede recibir el tráfico si hay una regla de NSG que permita el mismo tráfico que la regla NAT del equilibrador de carga.

¿Para qué podría crear reglas NAT? ¿Qué sucede si desea utilizar SSH o RDP para conectarse a una VM específica (y no utiliza Azure Bastion, que mencioné en el capítulo 2) o utiliza herramientas de administración para conectarse a un servidor de base de datos back-end? Si el equilibrador de carga distribuyera el tráfico a través de las VM de back-end, tendría que intentar conectarse una y otra vez, e incluso así podría no conectarse a la VM deseada.

Prácticas de seguridad recomendadas

Profundizaremos en algunos temas de seguridad en la parte 3 del libro, pero la seguridad debe ser una consideración permanente a medida que se compilan y ejecutan aplicaciones en Azure. La seguridad no debe ser un aspecto para agregar más adelante. Con el auge de la informática en la nube y las VM y aplicaciones web desechables, es fácil pasar por alto algunas prácticas recomendadas de seguridad básicas. Especialmente si trabaja en Azure como parte de una suscripción más amplia de la empresa, asegúrese de que cualquier recurso que cree no proporcione accidentalmente una manera para que los atacantes tengan acceso a su infraestructura.

¿Qué clase de cosas son malas? Bueno...algunas de las cosas que ya ha hecho en este libro. Los puertos de administración remota para SSH y RDP no deben abrirse a Internet público como lo ha hecho, o al menos se debe restringir el acceso si proviene de un rango de direcciones IP específico.

El procedimiento recomendado sería utilizar un servicio administrado, como Azure Bastion, o crear manualmente una VM segura que tenga disponible la administración remota. Según sea necesario, se conecta el host Azure Bastion o su única VM segura, y luego se conecta a través de la red virtual interna de Azure a las VM adicionales. Utilizó este enfoque de VM de servidor de salto básico en el capítulo 5. Este enfoque minimiza la superficie de ataque y reduce la necesidad de reglas NSG y reglas NAT del equilibrador de carga. En el capítulo 16 se describe Azure Security Center y se muestra cómo puede solicitar y abrir dinámicamente puertos de administración remota para un período específico, que es lo mejor de ambos mundos.

Incluso si trabaja con una suscripción privada de Azure que no tiene conectividad a otras suscripciones de Azure en la escuela o el trabajo, intente minimizar la cantidad de conectividad remota que proporciona.

Pruébelo ahora

Complete los siguientes pasos para crear una regla NAT del equilibrador de carga:

- 1 Abra Azure Portal y seleccione el icono Cloud Shell en la parte superior del panel.
- 2 Para crear una regla NAT de equilibrador de carga, defina un nombre, como natrulessh y el grupo IP de front-end a utilizar. La regla NAT examina el tráfico en un protocolo y puerto determinados, como el puerto TCP 50001. Cuando hay una coincidencia de reglas, el tráfico se reenvía al puerto 22 de back-end:

```
az network lb inbound-nat-rule create \
--resource-group azuremolchapter8 \
--lb-name loadbalancer \
--name natrulessh \
--protocol tcp \
--frontend-port 50001 \
--backend-port 22 \
--frontend-ip-name frontendpool
```

En este punto, ha creado un equilibrador de carga básico. Examine cómo se han reunido los componentes del equilibrador de carga:

```
az network lb show \
--resource-group azuremolchapter8 \
--name loadbalancer
```

Ha asignado una dirección IP pública al grupo IP de front-end y creó un sondeo de estado para comprobar el estado de una página de estado personalizada para un servidor web. Se creó una regla del equilibrador de carga para distribuir el tráfico web de sus clientes a un grupo de back-end; la regla utiliza sondeo de estado. También tiene una regla NAT de equilibrador de carga que permite el tráfico SSH, pero todavía no hay máquinas virtuales que reciban ese tráfico. Los clientes de su pizzería tienen hambre, así que vamos a crear algunas VM que puedan ejecutar su aplicación web y a la que el equilibrador de carga pueda distribuir el tráfico.

8.1.5 Asignación de grupos de VM a grupos de back-end

La sección final del equilibrador de carga define los grupos de back-end que incluyen una o más VM. Estos grupos de back-end contienen VM que ejecutan los mismos componentes de la aplicación, lo que permite al equilibrador de carga distribuir el tráfico a un grupo de back-end determinado y confiar en que cualquier VM de ese grupo puede responder correctamente a la solicitud del cliente. La figura 8.8 detalla cómo los grupos de back-end agrupan lógicamente las VM que ejecutan las mismas aplicaciones.

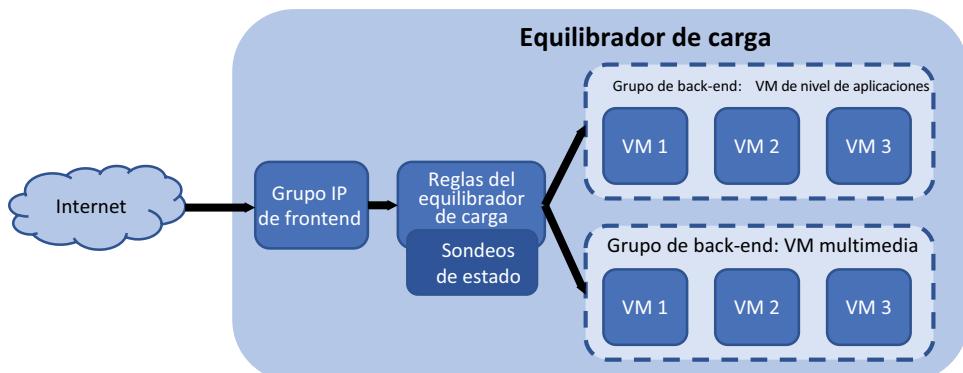


Figura 8.8 Se puede crear uno o más grupos de back-end en un equilibrador de carga. Cada grupo de back-end contiene una o más VM que ejecutan el mismo componente de aplicación. En este ejemplo, un grupo de back-end contiene VM que ejecutan el nivel de aplicación web, y otro grupo de back-end contiene las VM que sirven multimedia, como imágenes y video.

Usted crea y utiliza un equilibrador de carga con VM, pero todo funciona en el nivel de red virtual. El grupo IP de front-end utiliza direcciones IP públicas o privadas. El sondeo de estado analiza las respuestas en un puerto o protocolo determinados. Incluso cuando se utiliza un sondeo HTTP, el equilibrador de carga busca una respuesta de red positiva. Las reglas del equilibrador de carga se centran en cómo distribuir el tráfico desde un puerto externo en el grupo de front-end a un puerto en el grupo de back-end.

Cuando asigna VM al grupo de back-end que recibe tráfico distribuido por el equilibrador de carga, es la NIC virtual la que se conecta al equilibrador de carga. La VM se conecta a la NIC virtual. Vuelva a pensar en el capítulo 5; esta separación de VM y NIC virtual tiene sentido en términos de cómo se administran los recursos. Las reglas NSG controlan qué tráfico tiene permitido fluir a la VM, pero se aplican a una subred de red o a una NIC virtual, no a la VM.

¿Qué significa esto cuando se configuran grupos IP de back-end? Debe crear el resto de los recursos de la red virtual antes de poder conectar una VM al equilibrador de carga. Los pasos para crear los recursos de red deben ser un resumen de lo que aprendió hace unos capítulos, así que veamos cuánto recuerda.

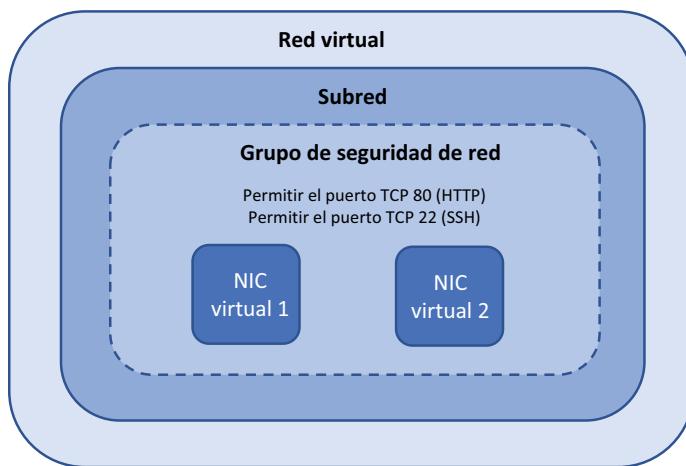


Figura 8.9 Para preparar la red virtual, en este ejercicio creará una red, una subred y NIC virtual protegidas por un NSG. Las reglas asociadas al NSG permiten el tráfico HTTP y SSH.

Pruébelo ahora

Complete los siguientes pasos para crear recursos de red adicionales, como se muestra en la figura 8.9:

- 1 Abra Azure Portal y seleccione el icono Cloud Shell en la parte superior del panel.
- 2 Cree una red y subred virtuales:

```

az network vnet create \
--resource-group azurermolchapter8 \
--name vnetmol \
--address-prefixes 10.0.0.0/16 \
--subnet-name subnetmol \
--subnet-prefix 10.0.1.0/24
  
```

En la práctica, es muy probable que estos recursos de red ya existan. Estos son también los mismos nombres y rangos de direcciones IP que usó en el capítulo 5. Debe limpiar los recursos Azure al final de cada capítulo, por lo que tal reutilización de rangos IP no debería ser un problema. Solo tenga en cuenta que normalmente no creará una red y subred virtuales cada vez que cree un equilibrador de carga; más bien, puede utilizar los recursos de red virtual existentes.

3 Cree un NSG:

```
az network nsg create \
--resource-group azuremolchapter8 \
--name webnsg
```

4 Cree una regla de NSG que permita que el tráfico desde el puerto TCP 80 llegue a las VM. Esta regla es necesaria para que las VM del servidor web reciban y respondan al tráfico de clientes:

```
az network nsg rule create \
--resource-group azuremolchapter8 \
--nsg-name webnsg \
--name allowhttp \
--priority 100 \
--protocol tcp \
--destination-port-range 80 \
--access allow
```

5 Agregue otra regla para permitir el tráfico SSH para administración remota. Esta regla de NSG funciona con la regla NAT del equilibrador de carga creada en la sección 8.1.4 para una de las VM:

```
az network nsg rule create \
--resource-group azuremolchapter8 \
--nsg-name webnsg \
--name allowssh \
--priority 101 \
--protocol tcp \
--destination-port-range 22 \
--access allow
```

6 Asocie el NSG con la subred creada en el paso 2. Las reglas NSG se aplican a todas las VM que se conectan a esta subred:

```
az network vnet subnet update \
--resource-group azuremolchapter8 \
--vnet-name vnetmol \
--name subnetmol \
--network-security-group webnsg
```

7 El equilibrador de carga funciona con NIC virtuales, así que cree dos NIC virtuales y conéctelas a la subred de red virtual. Especifique también el nombre del equilibrador de carga y el grupo de direcciones de back-end al que se conectan las NIC virtuales. La regla NAT del equilibrador de carga solo se conecta a esta primera NIC virtual creada:

```
az network nic create \
--resource-group azuremolchapter8 \
--name webnic1 \
--vnet-name vnetmol \
--subnet subnetmol \
--lb-name loadbalancer \
--lb-address-pools backendpool \
--lb-inbound-nat-rules natrulessh
```

- 8 Cree la segunda NIC de la misma manera, menos la regla NAT del equilibrador de carga:

```
az network nic create \
--resource-group azuremolchapter8 \
--name webnic2 \
--vnet-name vnetmol \
--subnet subnetmol \
--lb-name loadbalancer \
--lb-address-pools backendpool
```

8.2 Creación y configuración de VM con el equilibrador de carga

Hagamos una pausa para explorar lo que acabamos de crear. La figura 8.10 muestra el panorama general de cómo se ven los recursos de red y el equilibrador de carga. Observe la integración de estos recursos. El equilibrador de carga no puede existir por sí solo. Las NIC virtuales deben estar conectadas

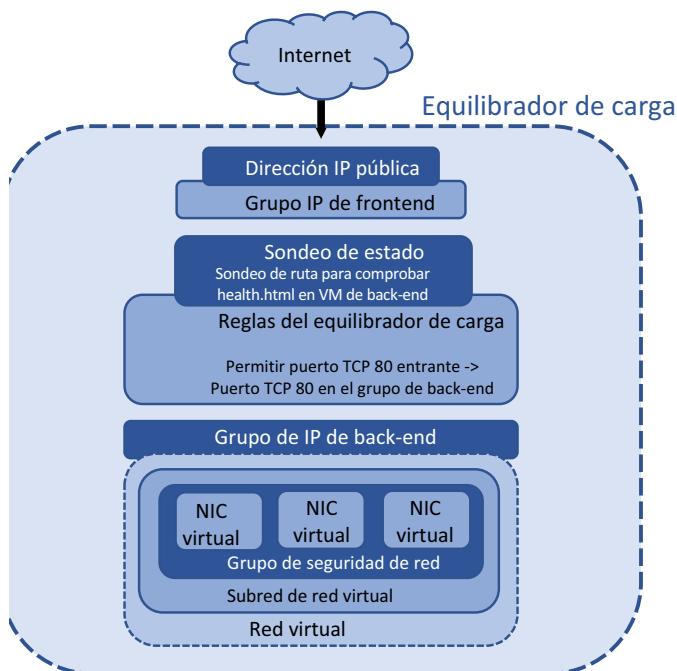


Figura 8.10 Aquí no se ha creado ninguna VM; la configuración del equilibrador de carga se ocupa de los recursos de red virtuales. Hay una estrecha relación entre el equilibrador de carga y los recursos de red virtual.

al equilibrador de carga para que se pueda distribuir cualquier tráfico. Esas NIC virtuales requieren una red y una subred virtual e, idealmente, están protegidas por un NSG. Aunque no lo crea, las VM que luego ejecutan la aplicación no tienen casi nada que ver con los pasos para crear y configurar el equilibrador de carga.

Ha creado muchos recursos de red y ha configurado varias partes del equilibrador de carga. La dirección IP pública y el equilibrador de carga se crearon en una zona de disponibilidad como recursos redundantes a nivel regional, así que vamos a crear dos VM en diferentes zonas para reforzar la forma en que las zonas de disponibilidad mejoran la alta disponibilidad de sus aplicaciones.

Si utiliza conjuntos de disponibilidad en lugar de zonas de disponibilidad, es aquí donde crea un conjunto de disponibilidad y, a continuación, agrega las VM; luego, la plataforma Azure distribuye las VM entre los dominios de error y de actualización. La idea es maximizar el uso de la alta disponibilidad de Azure para su pizzería, así que use zonas de disponibilidad.

Pruébelo ahora

Complete los siguientes pasos para crear las VM y conectarlas al equilibrador de carga:

- 1 Cree la primera VM y asígnela a una zona de disponibilidad con --zone 1:

```
az vm create \
--resource-group azuremolchapter8 \
--name webvm1 \
--image ubuntults \
--size Standard_B1ms \
--admin-username azuremol \
--generate-ssh-keys \
--zone 1 \
--nics webnic1
```

- 2 Cree la segunda VM y asígnela a la zona de disponibilidad 2; conecte la segunda NIC virtual que creó anteriormente utilizando --nics webnic2:

```
az vm create \
--resource-group azuremolchapter8 \
--name webvm2 \
--image ubuntults \
--size Standard_B1ms \
--admin-username azuremol \
--generate-ssh-keys \
--zone 2 \
--nics webnic2
```

Para ver el equilibrador de carga en acción, necesita instalar un servidor web básico, como lo hizo en el capítulo 2. También puede probar la regla NAT del equilibrador de carga. ¿Empieza a ver cómo todos estos componentes en Azure están relacionados y se complementan entre sí?

Pruébelo ahora

En el capítulo 5, analizamos el agente SSH. El agente SSH le permite pasar una clave SSH de una VM a la siguiente. Solo VM1 tiene una regla NAT del equilibrador de carga, así que debe usar el agente para conectarse a VM2. Complete los siguientes pasos para instalar un servidor web en las VM:

- 1 Inicie el agente SSH y agregue la clave SSH para que pueda conectarse a ambas VM:

```
eval $(ssh-agent) && ssh-add
```

- 2 Obtenga la dirección IP pública que está conectada al grupo IP de IP front-end del equilibrador de carga. Esta es la única manera de que el tráfico se dirija a través de las VM:

```
az network public-ip show \
    --resource-group azuremolchapter8 \
    --name publicip \
    --query ipAddress \
    --output tsv
```

- 3 Ya está listo para aplicar SSH a VM1. Especifique la dirección IP pública del equilibrador de carga (reemplace <your-ip-address> en el siguiente comando) y el puerto que se utilizó con la regla NAT del equilibrador de carga, como 50001. El parámetro -A utiliza el agente SSH para traspasar sus claves SSH:

```
ssh -A azuremol@<your-ip-address> -p 50001
```

En el capítulo 2, utilizó apt-get para instalar toda la pila LAMP, lo que incluye el servidor web Apache. Veamos algo un poco diferente del servidor web Apache con el servidor web independiente, pero de gran potencia, NGINX. Normalmente instalaría IIS en una VM Windows. Ejecute el siguiente comando para instalar el servidor web NGINX:

```
sudo apt update && sudo apt install -y nginx
```

- 4 En el repositorio de ejemplos de GitHub que utilizó en capítulos anteriores, hay una página web HTML básica y una página de comprobación de estado para el sondeo de estado del equilibrador de carga. Clone estos ejemplos en la VM:

```
git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
```

- 5 Copie la página HTML de ejemplo y la comprobación de estado en el directorio del servidor web:

```
sudo cp azure-mol-samples-2nd-ed/08/webvm1/* /var/www/html/
```

- 6 Ahora tiene que conectarse a la segunda VM e instalar el servidor web NGINX y el código de ejemplo. ¿Recuerda el agente SSH? Debería ser capaz de aplicar SSH de la VM 1 a la VM 2 en la dirección IP interna, privada:

```
ssh 10.0.1.5
```

7 Instale el servidor web NGINX:

```
sudo apt update && sudo apt install -y nginx
```

8 Clone los ejemplos de GitHub en la VM:

```
git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
```

9 Copie la página HTML de ejemplo y la comprobación de estado en el directorio del servidor web:

```
sudo cp azure-mol-samples-2nd-ed/08/webvm2/* /var/www/html/
```

Abra un navegador web y conéctese a la dirección IP pública del equilibrador de carga. Se carga la página web básica y muestra que su pizzería ahora tiene VM redundantes en zonas de disponibilidad que se ejecutan detrás de un equilibrador de carga, como se muestra en la figura 8.11. Podría necesitar actualizar forzosamente su navegador web para ver que tanto VM 1 como VM2 responden a medida que el equilibrador de cara distribuye el tráfico entre ellas.

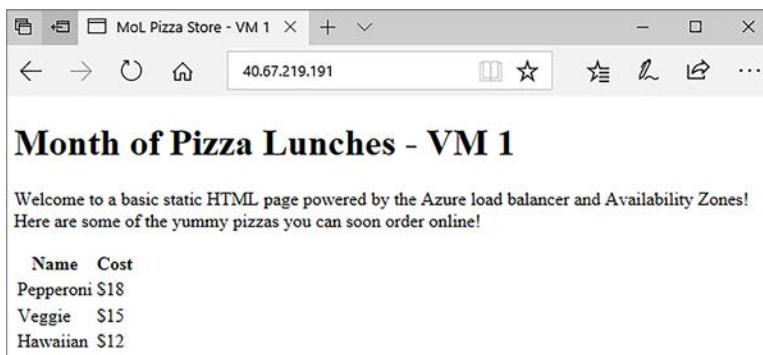


Figura 8.11 Al abrir la dirección IP pública del equilibrador de carga en un navegador web, el tráfico se distribuye a una de las VM que ejecutan su sitio web básico. El sondeo de estado del equilibrador de carga utiliza la página health.html para confirmar que el servidor web responde con un código HTTP 200 (OK). si es así, la VM está disponible como parte de la distribución de tráfico del equilibrador de carga.

8.3 Laboratorio: Visualización de plantillas de implementaciones existentes

Este capítulo une lo que aprendió en varios capítulos anteriores. Creó recursos de red, como en el capítulo 5. Hizo que el equilibrador de carga y las VM estuvieran altamente disponibles con zonas de disponibilidad, como en el capítulo 7. Y se instaló un servidor web e implementaron archivos de ejemplo, como en el capítulo 2. Su pizzería ha llegado muy lejos desde que partió como una página web básica en una sola VM, como lo era al comienzo del libro.

Para unir otro tema más de un capítulo anterior, en este laboratorio queremos que explore todos los recursos que componen el equilibrador de carga. Para ello, mire la plantilla de Resource Manager, como aprendió en el capítulo 6. El objetivo de este laboratorio es ver cómo una sola plantilla puede crear y configurar lo que ha tomado

muchas páginas y varios comandos de CLI. Y créame, ¡tomaría incluso más comandos de PowerShell! Siga estos pasos:

- 1 Abra Azure Portal.
- 2 Busque y seleccione Grupos de recursos en la barra de navegación del lado izquierdo del portal.
- 3 Elija el grupo de recursos, como azuremolchapter8.
- 4 Elija Exportar plantilla en la barra de la izquierda, como se muestra en la figura 8.12.
- 5 Para ver la parte relevante de la plantilla, seleccione cada uno de los recursos que se muestran en la lista. Tómese unos minutos para analizar esta plantilla y ver cómo están presentes todos los recursos y componentes que configuró en la CLI de Azure.

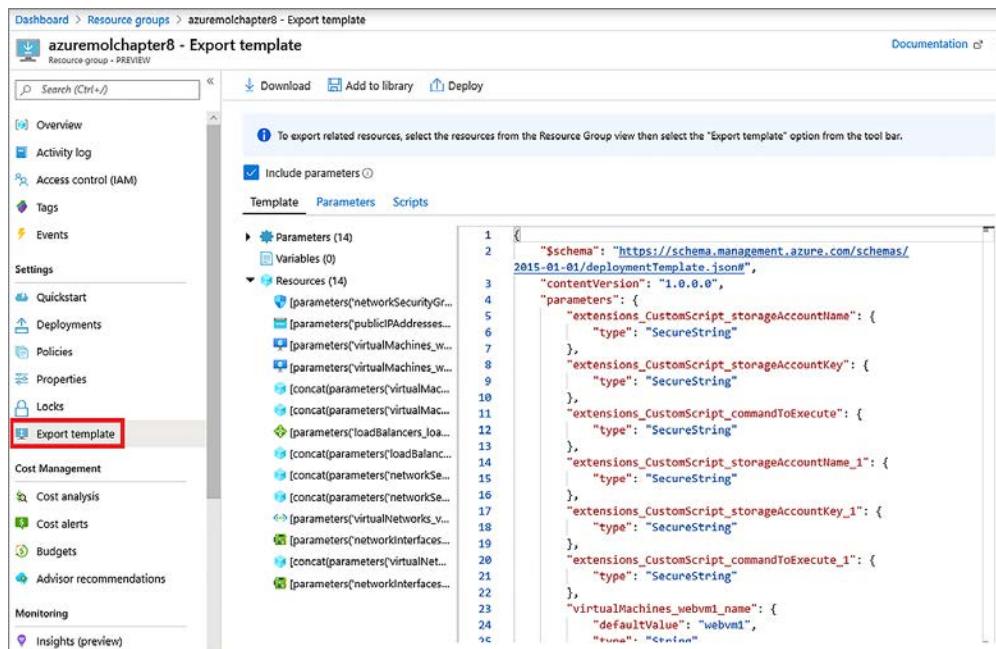


Figura 8.12 En Azure Portal, seleccione el grupo de recursos del equilibrador de carga y vea la plantilla Resource Manager.

Una plantilla hace que sea mucho más fácil implementar un entorno de aplicaciones altamente disponible, redundante y con equilibrio de carga. Puede cambiar el nombre, las reglas y el modo de distribución del equilibrador de carga, y dejar que la plantilla implemente y configure todo el entorno de la aplicación.

No olvide eliminar este grupo de recursos para aprovechar al máximo sus créditos Azure gratuitos.

Aplicaciones que escalan

En los dos capítulos anteriores, examinamos cómo compilar aplicaciones altamente disponibles y utilizar equilibradores de carga para distribuir el tráfico a varias VM que ejecutan su aplicación. Pero, ¿cómo ejecutar y administrar de forma eficiente varias VM y ejecutar el número correcto de instancias de VM cuando sus clientes más lo necesitan? Cuando la demanda del cliente aumenta, debe aumentar automáticamente el escalado de su aplicación para hacer frente a esa demanda. Y cuando la demanda disminuye, como en medio de la noche, cuando la mayoría de las personas duermen, quiere que la aplicación disminuya en escala y le ahorre algo de dinero.

En Azure, puede escalar automáticamente de forma horizontal los recursos de IaaS con conjuntos de escalado de máquinas virtuales. Estos conjuntos de escalado ejecutan VM idénticas, normalmente distribuidas detrás de un equilibrador de carga o Application Gateway. Usted define las reglas de escalado automático que aumentan o disminuyen el número de instancias de VM a medida que cambia la demanda del cliente. El equilibrador de carga o Application Gateway distribuye automáticamente el tráfico a las nuevas instancias de VM, lo que le permite centrarse en cómo compilar y ejecutar mejor sus aplicaciones. Los conjuntos de escalado le dan el control de los recursos IaaS con algunas de las ventajas elásticas de PaaS. Las aplicaciones web, que no cubrimos mucho en el último par de capítulos, ahora hacen una reaparición sólida con su propia capacidad de escalar con la demanda de la aplicación.

En este capítulo, examinaremos cómo diseñar y crear aplicaciones que puedan escalar automáticamente. Veremos por qué esta capacidad de escalar con la demanda le ayuda a ejecutar aplicaciones eficientes, y exploraremos diferentes maneras de escalar con base en diferentes métricas.

9.1 ¿Por qué compilar aplicaciones escalables y confiables?

¿Qué significa compilar aplicaciones que escalen? Le permite crecer y satisfacer la demanda de los clientes a medida que aumenta la carga de trabajo, incluso cuando está en el cine el fin de semana. Significa que no tiene que pagar una factura por

un montón de recursos adicionales que no utilizará o, incluso peor, que la aplicación deje de funcionar debido a la falta de recursos disponibles. El punto óptimo para las aplicaciones y los recursos que necesitan rara vez es estático. Generalmente, la aplicación exige una baja y un aumento del flujo durante el día y la noche, o entre los días laborables y los fines de semana.

Hay dos maneras principales de escalar los recursos, como se muestra en la figura 9.1: vertical y horizontalmente. Tanto los conjuntos de escalado de máquina virtual como las aplicaciones web pueden escalar vertical u horizontalmente.

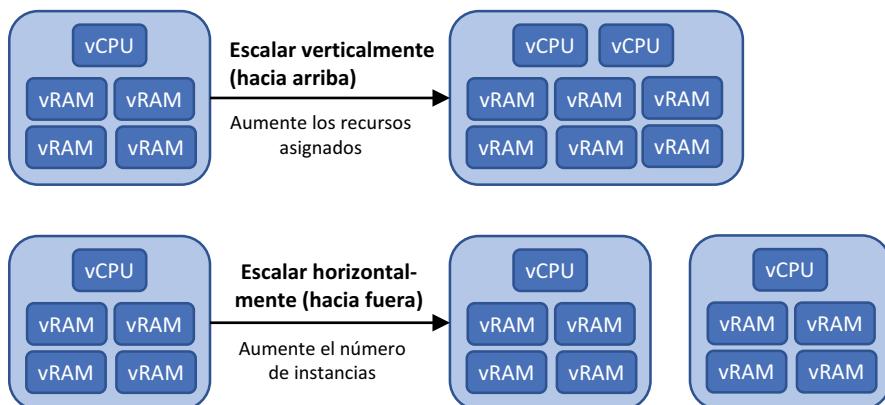


Figura 9.1 Puede escalar sus aplicaciones de forma horizontal o vertical. El método que utilice dependerá de cómo se compile la aplicación para manejar la escalabilidad. El escalado vertical ajusta los recursos asignados a una VM o a una aplicación web, como el número de núcleos de CPU o la cantidad de memoria. Este método para escalar una aplicación funciona bien si la aplicación ejecuta una sola instancia. El escalado horizontal cambia el número de instancias que ejecutan la aplicación y ayuda a aumentar la disponibilidad y la resiliencia.

Las aplicaciones escalables tienen una sólida relación con las aplicaciones altamente disponibles. En los capítulos 7 y 8 pasamos mucho tiempo con conjuntos de disponibilidad y zonas de disponibilidad, y cómo configurar equilibradores de carga. Ambos capítulos se centraron en la necesidad de ejecutar varias VM. Cuando la aplicación se puede escalar automáticamente, la disponibilidad de esa aplicación también se incrementa a medida que las VM se distribuyen entre los conjuntos de disponibilidad o las zonas de disponibilidad. Todo esto es algo bueno. El poder de Azure es que no es necesario preocuparse acerca de cómo agregar más instancias de aplicación, difundirlas en el hardware del centro de datos o incluso en centros de datos y luego actualizar los recursos de red para distribuir el tráfico a las nuevas instancias de la aplicación.

9.1.1 Escalabilidad vertical de las VM

La primera manera de escalar los recursos suele responder a lo que debió haber usado en el pasado. Si su aplicación comienza a funcionar de forma lenta a medida que más clientes la utilizan, ¿qué debería hacer normalmente? Aumentar la cantidad de CvPU o memoria, ¿verdad? Escala el recurso de forma *vertical* en respuesta a la demanda.

Uno de los usos más comunes del escalado vertical es para los servidores de bases de datos. Las bases de datos son voraces cuando se trata de procesos, ¡incluso más que los clientes de su pizzería! Los servidores de bases de datos suelen consumir todos los recursos proporcionados a una VM, incluso si no los utilizan inmediatamente. Esto puede dificultar la supervisión de las demandas reales en el sistema y saber cuándo es necesario escalar verticalmente y proporcionar más recursos. La figura 9.2 muestra la respuesta de escalado vertical típica para un servidor de base de datos que necesita más recursos.

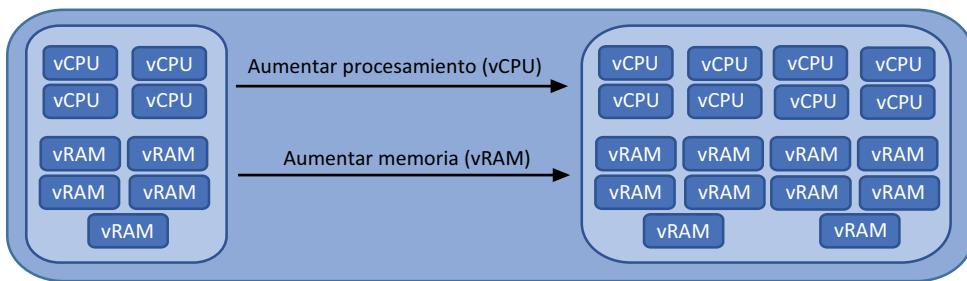


Figura 9.2 A medida que una base de datos crece, necesita más recursos para almacenar y procesar los datos in-memory. Para escalar verticalmente en este caso, agrega más CPU y memoria.

Es posible que necesite escalar más allá de la demanda de CPU o memoria. ¿Qué pasa si ejecuta un sitio web que contiene un montón de imágenes o video? Puede que no haya muchos requisitos de procesamiento, pero las demandas de ancho de banda pueden ser altas. Para aumentar el ancho de banda disponible, puede aumentar el número de NIC en su VM. Y si necesita almacenar más imágenes y video, agrega más almacenamiento. Puede agregar o eliminar recursos como NIC virtuales y almacenamiento a medida que la VM continúa funcionando.

REDIMENSIÓN DE MÁQUINAS VIRTUALES

En Azure, puede aumentar el tamaño de VM (escalar verticalmente) si necesita más procesos para su aplicación. En el capítulo 2, creó una VM básica. Su tamaño probablemente era algo así como Standard_D2s_v3. Ese nombre no le dice mucho acerca de los procesos asignados a una VM para determinar si es posible que necesite aumentar la CPU o la memoria. Si desea escalar verticalmente, necesita saber cuáles son sus opciones.

Pruébelo ahora

Siga adelante para ver los tamaños de VM y los procesos disponibles:

- 1 Abra Azure Portal en un navegador web y, a continuación, abra Cloud Shell.
- 2 Escriba el siguiente comando de la CLI de Azure para ver una lista de los tamaños de VM disponibles y los procesos que proporcionan:

```
az vm list-sizes --location eastus --output table
```

El resultado de `az vm list-sizes` varía de una región a otra y cambia con el tiempo a medida que Azure ajusta sus familias de VM. Este es un ejemplo condensado del resultado, que muestra `MemoryInMb` y `NumberOfCores` que proporciona cada tamaño de VM:

MaxDataDiskCount	MemoryInMb	Name	NumberOfCores
4	8192	Standard_D2s_v3	2
8	16384	Standard_D4s_v3	4
16	32768	Standard_D8s_v3	8
32	65536	Standard_D16s_v3	16
8	4096	Standard_F2s_v2	2
16	8192	Standard_F4s_v2	4
32	16384	Standard_F8s_v2	8
2	2048	Standard_B1ms	1
2	1024	Standard_B1s	1
4	8192	Standard_B2ms	2
4	4096	Standard_B2s	2

Así, su VM `Standard_D2s_v3` proporciona dos núcleos de CPU y 8 GB de memoria, más que suficiente para una VM básica que ejecute un servidor web. Supongamos que su pizzería en línea comienza a recibir algunos pedidos y desea escalar verticalmente. Puede utilizar `az vm resize` para elegir otro tamaño. Especifica el tamaño de VM que tiene el número de núcleos de CPU y la memoria que necesita su aplicación.

La CPU y la memoria adicionales no aparecen mágicamente en la VM. Este comportamiento puede ser un poco diferente de lo que experimenta con Hyper-V o VMware en el ámbito local. Dentro de lo razonable, puede agregar o quitar procesos básicos en un entorno local a medida que la VM continúa funcionando. En Azure, normalmente se requiere un reinicio de una VM cuando cambia el tamaño para registrar los nuevos procesos y activar las reglas de facturación apropiadas. Cuando desee escalar verticalmente, prepárese para un tiempo de inactividad mientras se reinicia la VM.

REDUCCIÓN VERTICAL

¿Qué sucede si tiene una VM con más recursos de los que necesita? Esta situación suele ser más común que una VM que tiene menos recursos de los necesarios. Los dueños de aplicaciones pueden elegir un tamaño de VM mayor que el requerido, para asegurarse de que la aplicación funcione sin problemas. Todos esos recursos desperdiciados cuestan dinero, y es fácil que los costos pasen inadvertidos hasta que llega la cuenta a fin de mes.

La capacidad de escalar recursos funciona en ambas direcciones. Nos hemos centrado en cómo escalar *verticalmente* los recursos, pero funcionan los mismos conceptos para *reducir* los recursos. Es importante identificar los tamaños de VM en uso y la cantidad de demanda que las aplicaciones hacen en esos recursos. A continuación, puede utilizar `az vm resize` para seleccionar un tamaño de VM con menos núcleos de CPU y memoria. Actualmente se necesita un reinicio de VM para cualquier operación de redimensionamiento.

9.1.2 Escalado vertical de aplicaciones web

Las aplicaciones web pueden escalarse vertical u horizontalmente basándose en las necesidades de los recursos, de la misma manera que las VM. Cuando creó una aplicación web en el capítulo 3, el tamaño estándar S1 predeterminado proporcionó

un núcleo de CPU y 1,75 GB de RAM. Cada nivel de aplicación web y tamaño proporciona una cantidad establecida de recursos como núcleos de CPU, memoria y espacios de ensayo. Incluso si cambia el tamaño o la asignación de recursos predeterminada, o si elige un tamaño de aplicación web diferente, el concepto sigue siendo el mismo.

Si crea su aplicación web y encuentra que la aplicación requiere más recursos de los que proporciona el plan de servicio, puede cambiar a un nivel diferente, como se muestra en la figura 9.3. El mismo proceso funciona si tiene más recursos que los que necesita. Su aplicación web puede escalar o reducirse manualmente de esta manera según sea necesario.

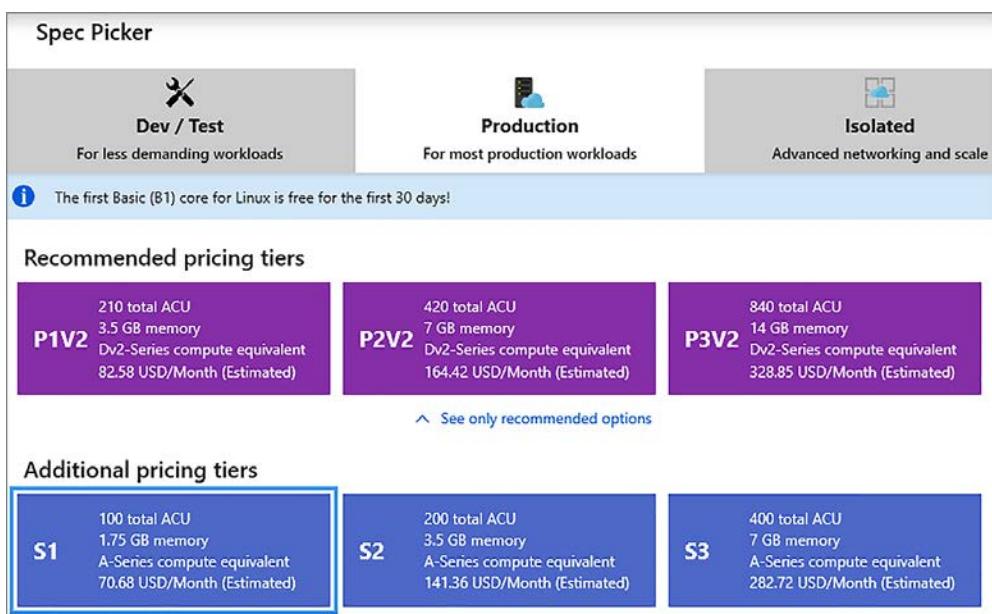


Figura 9.3 Para escalar manualmente una aplicación web verticalmente, se cambia el nivel de precios (tamaño) del plan de servicio subyacente de la aplicación. El plan de servicio de la aplicación define la cantidad de recursos asignados a la aplicación web. Si la aplicación requiere una cantidad diferente de almacenamiento, número de CPU o ranuras de implementación, puede cambiar a un nivel diferente para ajustar los recursos asignados a la demanda de la aplicación.

9.1.3 Escalado horizontal de recursos

Un enfoque diferente para satisfacer la demanda es escalar horizontalmente. Para escalar verticalmente, debe aumentar la cantidad de CPU y memoria asignada a un único recurso, como una VM. En cambio, para escalar horizontalmente, debe aumentar el número de VM, como se muestra en la figura 9.4.

Para escalar horizontalmente, su aplicación necesita estar al tanto de esta capacidad y ser capaz de procesar datos sin conflictos. Una aplicación web es un gran candidato para escalar horizontalmente, ya que la aplicación normalmente puede procesar los datos por sí misma.

A medida que se compilan aplicaciones más complejas, se puede dividir una aplicación en componentes individuales más pequeños. Si vuelve a pensar en las colas de Azure Storage del capítulo 4, puede

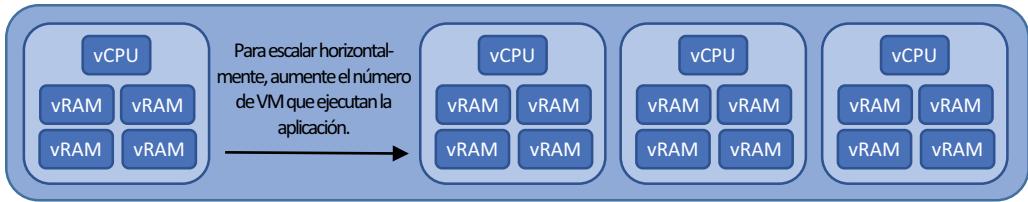


Figura 9.4 Para hacer frente a un aumento de la demanda en su aplicación, puede aumentar el número de máquinas virtuales que ejecutan la aplicación, distribuyendo la carga entre múltiples máquinas virtuales en lugar de máquinas virtuales de una sola instancia cada vez más grandes.

tener un componente de aplicación que reciba los pedidos web de front-end y otro componente de aplicación que procese esos pedidos y los transmita a la pizzería. El uso de colas de mensaje es un enfoque para diseñar y escribir aplicaciones que pueden funcionar en un entorno que escala horizontalmente. Este enfoque también le permite escalar cada componente de aplicación por separado y utilizar diferentes tamaños de VM o planes de aplicación web para maximizar la eficiencia y reducir sus gastos mensuales.

Históricamente, se escalaba de manera vertical porque era más fácil agregar más procesos a una aplicación y esperar que funcionara bien. Configurar un clúster de recursos y escalar una aplicación horizontalmente, solía ser complejo en el mundo físico. Con la informática en la nube y la virtualización, los desafíos de escalar horizontalmente se minimizan hasta el punto de que a menudo se puede escalar de forma horizontal con más rapidez que de forma vertical y sin tiempos de inactividad.

¿Recuerda el comando `az vm resize` antes mencionado en este capítulo? ¿Qué sucede cuando se completa la operación de redimensionamiento de la VM? Se reinicia la VM. Si esa es la única instancia de su aplicación, nadie puede acceder a ella hasta que vuelva a estar en línea. Cuando se escala horizontalmente, no hay tiempo de inactividad cuando se agregan instancias de VM: cuando las nuevas VM están listas, empiezan a procesar algunas de las solicitudes de la aplicación. Los sondeos de estado de los equilibradores de carga (capítulo 8) detectan automáticamente cuando una nueva VM en el grupo de back-end está lista para procesar las solicitudes de los clientes y el tráfico comienza a distribuirse a ella.

Azure está diseñado para darle flexibilidad y capacidad de elección cuando se trata de cómo escalar. Si está diseñando un entorno de aplicación nuevo, le sugerimos que implemente un enfoque de escalado horizontal. Las VM tienen un recurso utilísimo en Azure que puede ayudarle con esto: los conjuntos de escalado de las máquinas virtuales.

9.2 Conjuntos de escalado de máquinas virtuales

Las VM son algunas de las cargas de trabajo más comunes en Azure, por una buena razón. La curva de aprendizaje para compilar y ejecutar una VM es simple, porque la mayoría de lo que ya sabe se transfiere directamente a Azure. Los servidores web están entre las cargas de trabajo más comunes para una VM, lo que es conveniente ya que no tiene que aprender nuevas habilidades para transferir su conocimiento sobre cómo ejecutar Apache, IIS o NGINX en una VM de Azure.

¿Qué pasa con un clúster de VM que ejecuta un servidor web? ¿Cómo se encargaría de eso en su entorno local normal? Para empezar, hay muchas soluciones de clúster posibles. ¿Qué pasa con las actualizaciones de sus servidores físicos o VM? ¿Cómo se encargaría de estas? ¿Qué ocurriría si deseara aumentar o disminuir automáticamente el número de instancias del clúster? ¿Necesita utilizar otra herramienta? En la figura 9.5 se muestra un esquema de un conjunto de escalado de máquina virtual.

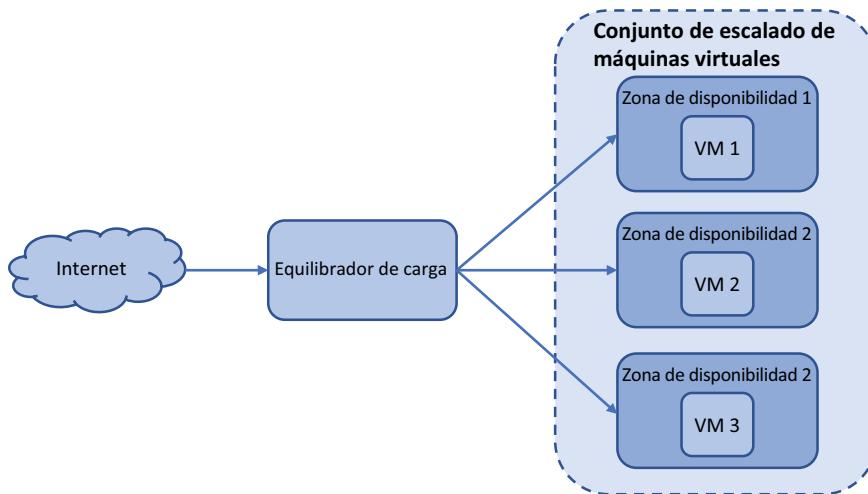


Figura 9.5 Un conjunto de escalado de máquina virtual agrupa lógicamente un conjunto de VM. Las VM son idénticas y pueden administrarse, actualizarse y escalarse centralmente. Puede definir métricas que aumenten o disminuyan automáticamente el número de VM en el conjunto de escalado basándose en la carga de la aplicación.

Un conjunto de escalado simplifica la forma de ejecutar y administrar varias VM para proporcionar una aplicación altamente disponible y con equilibrio de carga. Debe indicarle a Azure qué tamaño de VM usar, una imagen base para la VM, y cuántas instancias quiere. A continuación, puede definir las métricas de la CPU o de la memoria para aumentar o disminuir automáticamente el número de instancias en respuesta a la carga de la aplicación, o de forma programada a horas punta de clientes. Los conjuntos de escalado combinan el modelo IaaS de las VM con el poder de funcionalidades PaaS como escalado, redundancia, automatización y administración centralizada de recursos.

¿Un solo conjunto de escalado de VM?

Si compila aplicaciones en VM, planifique comenzar con un conjunto de escalado, incluso si solo necesita una VM. ¿Por qué? Un conjunto de escalado puede expandirse en cualquier momento, y crea automáticamente las conexiones a un equilibrador de carga o a una gateway de aplicaciones. Si la demanda de la aplicación aumenta repentinamente en dos meses, puede decirle al conjunto de escalado que cree una o dos instancias de VM adicionales.

Para expandir una VM normal, independiente, necesita agregar esa VM a un equilibrador de carga; y si no comenzó con la VM en un conjunto de disponibilidad o zona de disponibilidad, tiene que planificar el modo de hacer que esas VM estén altamente disponibles. Al crear un conjunto de escalado desde el comienzo, incluso para una VM, prepara su aplicación para el futuro con un mínimo de trabajo adicional requerido.

9.2.1 Creación de un conjunto de escalado de máquina virtual

Aunque un conjunto de escalado hace que sea más sencillo compilar y ejecutar aplicaciones altamente disponibles, es necesario crear y configurar algunos componentes nuevos. Dicho esto, para implementar una escala establecida con la CLI de Azure, el proceso se reduce a dos comandos.

Pruébelo ahora

Complete los siguientes pasos para crear un conjunto de escalado con la CLI de Azure:

- 1 Abra Azure Portal y seleccione el icono Cloud Shell en la parte superior del panel.
- 2 Cree un grupo de recursos con `az group create`; especifique un nombre para el grupo de recursos, como `azuremolchapter9` y una ubicación:

```
az group create --name azuremolchapter9 --location westeurope
```

Los conjuntos de escalado pueden utilizar zonas de disponibilidad para seleccionar una región compatible.

- 3 Para crear un conjunto de escalado, especifique el número de instancias de VM que desee y la forma en que las instancias de VM deben gestionar las actualizaciones de su configuración. Cuando realiza un cambio en las VM, por ejemplo, como instalar una aplicación o aplicar actualizaciones del sistema operativo invitado, las VM pueden actualizarse automáticamente en cuanto detecten el cambio. O bien, puede establecer la directiva de actualización manual y aplicar las actualizaciones en el momento que más le acomode. El resto de los parámetros debieran serle familiares de cuando creó una sola VM:

```
az vmss create \
--resource-group azuremolchapter9 \
--name scalesetmol \
--image UbuntuLTS \
--admin-username azuremol \
--generate-ssh-keys \
--instance-count 2 \
--vm-sku Standard_B1ms \
--upgrade-policy-mode automatic \
--lb-sku standard \
--zones 1 2 3
```

¡Listo! Ha creado varias VM en una zona de disponibilidad que puede escalar. Prepárese para algo realmente genial sobre el conjunto de escalado que acaba de crear con la CLI de Azure. ¿Recuerda todo el capítulo sobre equilibradores de carga (capítulo 8), todos los comandos CLI que tenía que utilizar y cómo miraba las plantillas para simplificar la forma de crear un equilibrador de carga? El comando `az vmss create` creó y configuró un equilibrador de carga automáticamente.

Recuerde sus límites de cuotas

Mencioné este problema con la cuota en el capítulo 7, pero vale la pena repetirlo en caso de que tenga problemas. En Azure, las cuotas predeterminadas de su suscripción le impiden implementar accidentalmente recursos y olvidarse de ellos, lo que le costaría dinero. Puede ver una lista de cuotas en <http://mng.bz/ddcx>.

Puede tener problemas con sus cuotas cuando crea varias VM, así como también si no elimina los recursos de los capítulos y ejercicios anteriores. Si ve texto de error en las líneas de

```
Operation results in exceeding quota limits of Core.  
Maximum allowed: 4, Current in use: 4, Additional requested: 2.
```

es suficiente indicación de que necesita solicitar un aumento de sus cuotas. Puede ver su cuota actual para una región determinada de la siguiente manera:

```
az vm list-usage --location westeurope
```

Para solicitar un aumento de sus cuotas para una región, siga los pasos descritos en <http://mng.bz/Xq2f>.

La CLI de Azure le ayuda a crear un conjunto de escalado con un mínimo de notificaciones. Se ha creado y configurado un equilibrador de carga, se ha asignado una dirección IP pública y se han agregado las instancias del conjunto de escalado de VM al grupo IP de back-end.

Pruébelo ahora

Revise los recursos creados con su conjunto de escalado, como se describe a continuación.

Para ver qué recursos se crearon con el conjunto de escalado, ejecute el siguiente comando:

```
az resource list \  
--resource-group azuremolchapter9 \  
--output table
```

El resultado es similar al siguiente ejemplo. Mire la columna Tipo para comprobar que se crearon una red virtual, una dirección IP pública y un equilibrador de carga:

Nombre	ResourceGroup	Type
mol	azuremolchapter9	Microsoft.Compute/virtualMachineScaleSets
molLB	azuremolchapter9	Microsoft.Network/loadBalancers
molLBIP	azuremolchapter9	Microsoft.Network/publicIPAddresses
molVNET	azuremolchapter9	Microsoft.Network/virtualNetworks

¿Qué significa toda esta magia? Cuando crea un conjunto de escalado con la CLI de Azure, se crea un equilibrador de carga con redundancia a nivel regional y una dirección IP pública. Las VM se crean y se agregan a un grupo IP de back-end en el equilibrador de carga. Se crean reglas NAT que le permiten conectarse a las instancias

de VM. Lo único que falta son las reglas de equilibrio de carga, ya que varían en función de las aplicaciones que deseé ejecutar. A medida que agrega o quita VM al conjunto de escalado, la configuración del equilibrador de carga se actualiza automáticamente para permitir que el tráfico se distribuya a las nuevas instancias. Esta magia no se limita a la CLI de Azure; si usa Azure PowerShell o Azure Portal, estos recursos de red de apoyo se crean y conectan para trabajar en conjunto.

Pruébelo ahora

El escalado se creó con dos instancias. Puede escalar manualmente el número de instancias de VM en el conjunto de escalado. Al hacerlo, el equilibrador de carga actualiza automáticamente la configuración del grupo IP de back-end. Defina la nueva capacidad `--new-capacity` del conjunto de escalado en cuatro instancias de la siguiente manera:

```
az vmss scale \
--resource-group azuremolchapter9 \
--name scalesetmol \
--new-capacity 4
```

9.2.2 Creación de reglas de escalado automático

Cuando creó el conjunto de escalado, implementó un número fijo de instancias. Una de las funciones más importantes de los conjuntos de escalado es la capacidad de escalar automáticamente de forma horizontal el número de instancias de VM que ejecuta el conjunto de escalado.

Como se muestra en la figura 9.6, el número de instancias de un conjunto de escalado puede aumentar automáticamente a medida que aumenta la carga de la aplicación. Piense en una aplicación comercial típica en su entorno. A principios de la jornada laboral, los usuarios comienzan a acceder a la aplicación, lo que hace que aumente la carga de recursos en esas instancias de VM. Para garantizar un rendimiento óptimo de la aplicación, el conjunto de escalado agrega automáticamente más instancias de VM. El equilibrador de carga comienza a distribuir automáticamente el tráfico a las nuevas instancias. Más tarde en el día laboral, cuando los usuarios se van a casa, cae la demanda de la aplicación. Las instancias de VM utilizan menos recursos, por lo que el conjunto de escalado elimina automáticamente algunas instancias de VM para reducir los recursos innecesarios y disminuir los costos.

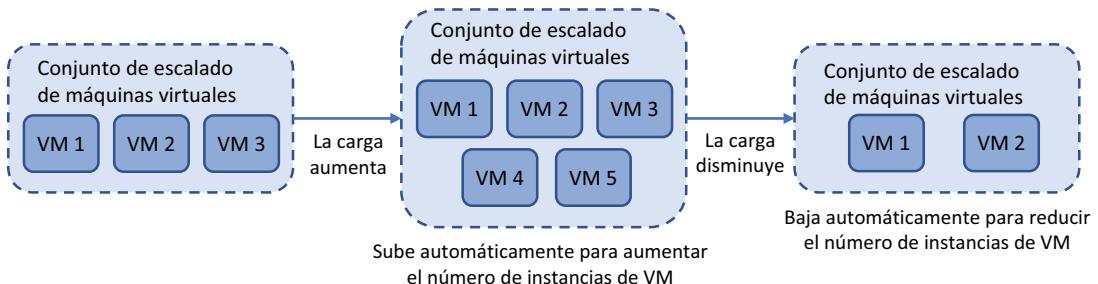


Figura 9.6 Los conjuntos de escalado se pueden reducir o escalar automáticamente de forma horizontal. Usted define reglas para supervisar ciertas métricas que activan las reglas para aumentar o disminuir el número de instancias de VM que se ejecutan. A medida que la demanda de la aplicación cambia, también lo hace el número de instancias de VM. Este enfoque maximiza el rendimiento y la disponibilidad de su aplicación, al mismo tiempo que minimiza los cargos innecesarios cuando disminuye la carga de la aplicación.

Puede basar las reglas de ajuste de escalado en varias métricas. Puede ver las métricas de host para consumo básico de recursos, configurar la recolección de métricas de VM dentro del invitado para analizar los contadores de rendimiento específicos de la aplicación o utilizar Azure Application Insights para supervisar en profundidad el código de la aplicación.

También usa programaciones para definir cierto número de instancias de VM en un conjunto de escalado para un período específico. En el ejemplo de una aplicación empresarial común para la que la demanda es mayor durante las horas de trabajo que por la tarde, es posible que desee definir un número fijo mayor de instancias para ejecutar durante las horas de trabajo y definir un número fijo de instancias para ejecutarse por la noche.

Las reglas de escalado automático basadas en métricas supervisan el rendimiento durante un intervalo de tiempo definido, 5 minutos, por ejemplo, y pueden tardar unos pocos minutos en activar las nuevas instancias de VM y configurarlas para el uso de las aplicaciones. Si utiliza horarios fijos para ajustar el escalado automático del número de instancias de VM en el conjunto de escalado, esos recursos adicionales ya están en uso y el equilibrador de carga distribuye el tráfico durante todo el día.

El uso de horarios requiere una base para la demanda típica de la aplicación y no toma en cuenta una demanda mayor o menor en ciertas partes de la cuenta comercial o del ciclo de ventas. Puede terminar con más recursos de los que realmente necesita, pagando más de lo necesario. Y es posible que haya situaciones en las que la carga de la aplicación sea mayor que el número de instancias de VM que el conjunto de escalado puede proporcionar.

Pruébelo ahora

Complete los siguientes pasos para crear reglas de escalado automático para un conjunto de escalado:

- 1 Busque y seleccione Grupos de recursos en la barra de navegación del lado izquierdo de Azure Portal.
- 2 Elija el grupo de recursos que creó para la implementación de la plantilla, como azuremolchapter9.
- 3 Seleccione el conjunto de escalado en la lista de recursos, como scalesetmol.
- 4 Debajo de Configuración, en la parte izquierda de la ventana Conjunto de escalado, seleccione Escalado. Puede escalar manualmente o crear sus propias reglas de autoescalado.
- 5 Elija crear reglas de autoescalado personalizadas.
- 6 Escriba un nombre, como autoscale y, a continuación, defina un conteo de instancias mínimo, máximo y predeterminado. Para este ejercicio, fije el mínimo en 2, el máximo en 10, y el predeterminado en 2.
- 7 Elija agregar una regla y, a continuación, revise la configuración de reglas disponible, como se muestra en la figura 9.7.

Los parámetros predeterminados buscan el consumo promedio de la CPU. La regla se activa cuando la carga es superior al 70 % durante un intervalo de 10 minutos. El conjunto de escalado se incrementa en 1 instancia de VM y las reglas luego esperan 5 minutos antes de que comiencen a supervisar y se pueda activar la siguiente regla.

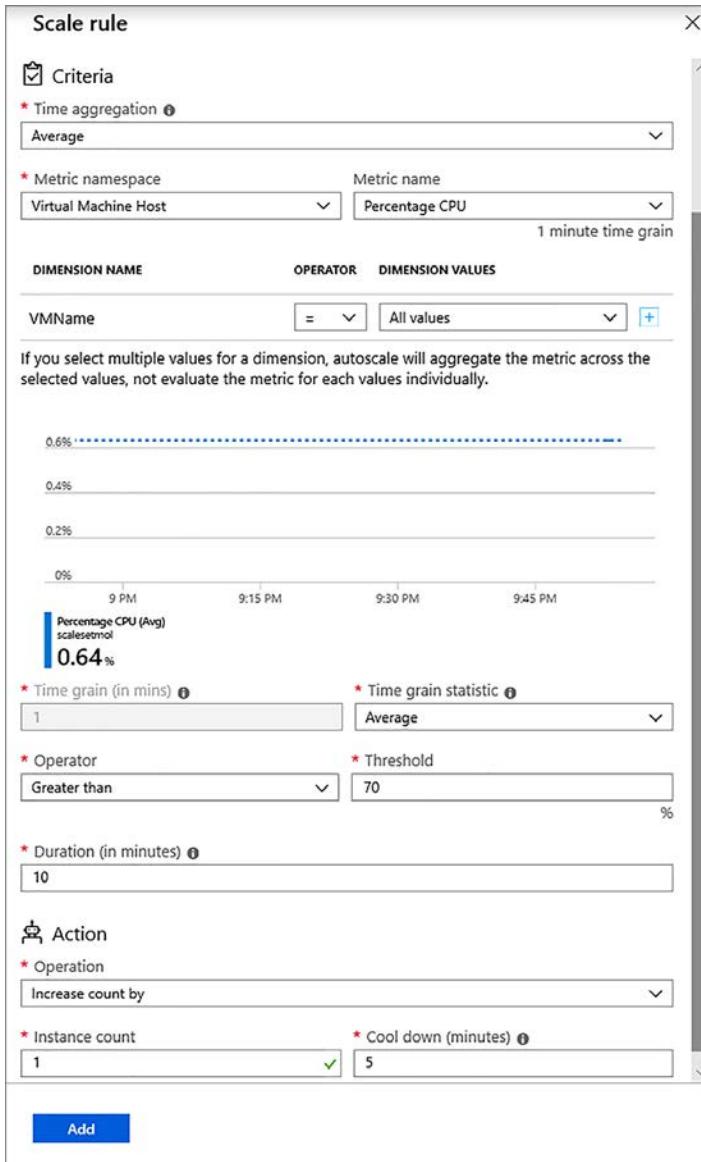


Figura 9.7 Cuando se agrega una regla de escalado automático, se define el comportamiento exacto requerido para que la regla se active.

Este período de enfriamiento proporciona a las nuevas instancias de VM tiempo para implementar y comenzar a recibir el tráfico del equilibrador de carga, lo que debería disminuir la carga global de la aplicación en el conjunto de escalado. Sin este período de enfriamiento, las reglas pueden activar otra instancia de VM que se agregaría antes de que la carga haya empezado a distribuirse en la instancia de VM anterior que se creó.

- 8 Para crear la regla, seleccione Agregar.
- 9 Elija agregar otra regla. Esta vez, configure la regla para disminuir el conteo en uno cuando la carga promedio de la CPU sea inferior al 30 % en un plazo de 5 minutos.
- 10 Revise sus reglas, como se muestra en la figura 9.8, y luego seleccione Guardar.

The screenshot shows the Azure portal's Autoscale settings interface. At the top, there are buttons for Save (highlighted in red), Discard, Disable autoscale, and Refresh. Below that, tabs for Configure, Run history, JSON, and Notify are visible. The main area is titled 'Default Auto created scale condition'. It displays a warning: 'Delete warning: The very last or default recurrence rule cannot be deleted. Instead, you can disable autoscale to turn off autoscale.' Under 'Scale mode', the option 'Scale based on a metric' is selected. The 'Rules' section contains two entries:

- Scale out:** When 'scalesetmol' (Average) Percentage CPU > 70, Increase instance count by 1.
- Scale in:** When 'scalesetmol' (Average) Percentage CPU < 30, Decrease instance count by 1.

Below the rules, there's a link '+ Add a rule'. The 'Instance limits' section shows 'Minimum' set to 2, 'Maximum' set to 10, and 'Default' set to 2. A note at the bottom states: 'This scale condition is executed when none of the other scale condition(s) match'. At the bottom of the page, there's a link '+ Add a scale condition'.

Figura 9.8 Ahora debe tener una regla que aumente el conteo de instancias en 1 cuando la carga promedio de la CPU sea superior al 70 %, y otra regla que disminuya el conteo de instancias en uno cuando la carga promedio de la CPU sea inferior al 30 %.

También puede configurar reglas de escalado automático con la CLI de Azure, Azure PowerShell o en plantillas. El portal proporciona una cómoda visualización para revisar las reglas y ver las opciones disponibles para cada parámetro. A medida que se crean reglas más complejas, las plantillas proporcionan una forma de crear conjuntos de escalado con el mismo conjunto de reglas de forma reproducible.

9.3 ¿Desea escalar una aplicación web?

Si le interesaron las aplicaciones web en el capítulo 3, o en las tablas y colas de Azure en el capítulo 4, la cantidad de contenido de VM de IaaS de estos tres últimos capítulos pueden haberlo dejado agotado. ¿Acaso la nube no era supuestamente más fácil que esto? Para componentes PaaS como Web Apps, ¡absolutamente!

No queremos que se complique con la forma de proporcionar la misma alta disponibilidad y capacidades de escalado automático a las aplicaciones web en las próximas páginas. ¡La verdad es que es mucho más fácil de hacer! Al igual que la mayoría de las cosas, la elección entre IaaS y PaaS es un equilibrio entre flexibilidad y facilidad de manejo. Gran parte de la redundancia subyacente se abstrae en servicios PaaS como Web Apps, así que no necesita un capítulo completo sobre alta disponibilidad y otro capítulo sobre equilibradores de carga.

El camino de IaaS para compilar y ejecutar sus propias VM o conjuntos de escalado con equilibradores de carga y zonas de disponibilidad puede originarse de una necesidad comercial o una restricción. Es posible que los desarrolladores, los ingenieros de operaciones o las herramientas y los flujos de trabajo no estén listos para trabajar de manera conjunta en Web Apps. Dicho esto, sugerimos encarecidamente que busque si Web Apps tiene nuevas implementaciones de aplicaciones. El uso de componentes PaaS como Web Apps le da más tiempo para centrarse en las aplicaciones y en sus clientes en lugar de la infraestructura y la administración.

Pruébelo ahora

Complete los siguientes pasos para crear una aplicación web con la CLI de Azure:

- 1 En el capítulo 3, creó una aplicación web en Azure Portal. Al igual que con la mayoría de los recursos, a menudo es más rápido y más fácil de usar la CLI de Azure. En Azure Portal, abra Cloud Shell.
- 2 Cree un plan de App Services de un tamaño estándar S1. Este tamaño le permite escalar automáticamente hasta 10 instancias de su aplicación web:

```
az appservice plan create \
  --name appservicemol \
  --resource-group azuremolchapter9 \
  --sku s1
```

- 3 Cree una aplicación web que utilice un repositorio Git local para la implementación, como lo hizo en el capítulo 3:

```
az webapp create \
  --name webappmol \
  --resource-group azuremolchapter9 \
  --plan appservicemol \
  --deployment-local-git
```

Todos los conceptos y casos en torno a las reglas de escalado automático y las programaciones de escalado analizadas en la sección 9.2.2 también se aplican a las aplicaciones web. A modo de resumen rápido, estos son un par de casos comunes para escalar automáticamente las aplicaciones web:

- Aumente o disminuya automáticamente el número de instancias de aplicaciones web con base en métricas de rendimiento, a fin de satisfacer la demanda de las aplicaciones durante toda la jornada laboral.
- Programe una aplicación web para aumentar automáticamente el número de instancias al inicio del día laboral y, a continuación, disminuir el número de instancias al finalizar la jornada.

En el caso de la pizzería, la aplicación web puede recibir un mayor tráfico al final del día y durante la noche, por lo que no hay un conjunto de reglas de escalado automático que se aplique a cada situación. Nuevamente, es necesario tener una base de referencia sobre el rendimiento de la aplicación para entender cómo se ejecuta bajo uso normal y la métrica de rendimiento en la cual necesita escalar horizontalmente. Incluso cuando utiliza programación de escalado automático, debe seguir monitoreando y rastreando las subidas de demanda de su aplicación para crear reglas que sean compatibles con ese patrón de uso.

Pruébelo ahora

Complete los siguientes pasos para crear reglas de escalado automático para aplicaciones web:

- 1 Busque y seleccione Grupos de recursos en la barra de navegación del lado izquierdo de Azure Portal.
- 2 Elija el grupo de recursos que creó para la implementación de la aplicación web, como azuremolchapter9.
- 3 Seleccione la aplicación web de la lista de recursos, como webappmol.
- 4 Debajo de Configuración a la izquierda en la ventana de aplicaciones web, seleccione Escalar horizontalmente (App Services Plan).
- 5 De nuevo, elija configurar reglas de autoescalamiento personalizadas, no solo escalar manualmente la aplicación web.
- 6 Escriba un nombre, como autoscalewebapp y, a continuación, defina un conteo de instancias mínimo, máximo y predeterminado. Para este ejercicio, fije el mínimo en 2, el máximo en 5, y el predeterminado en 2.
- 7 Seleccione para agregar una regla y, a continuación, revise la configuración de reglas disponible. Esta ventana se ve igual que las reglas de escalado para conjuntos de escalado. Los parámetros predeterminados observan el consumo promedio de la CPU y se activan cuando la carga supera el 70 % durante un intervalo de 10 minutos. La aplicación web se incrementa en una instancia de VM y luego las reglas esperan 5 minutos antes de comenzar a supervisar y se pueda activar la siguiente regla.
- 8 Elija agregar otra regla. Esta vez, configure la regla para disminuir el conteo en uno cuando la carga promedio de la CPU sea inferior al 30 % en un plazo de 5 minutos.
- 9 Revise y guarde sus reglas.

Cuando las reglas de escalado automático activan la aplicación web para escalado horizontal, la plataforma Azure actualiza la distribución de tráfico a las instancias de aplicación web disponibles. No hay ningún equilibrador de carga expuesto, como ocurre con los conjuntos de escalado, pero el tráfico se sigue distribuyendo automáticamente entre las instancias de la aplicación web a medida que el entorno se amplía o reduce. El concepto es similar, solo que abstraído de usted, porque se supone que debe disfrutar del enfoque PaaS y no preocuparse tanto.

Tanto los conjuntos de escalado como las aplicaciones web proporcionan una forma de crear reglas que escalan automáticamente el número de instancias que ejecutan las aplicaciones. Con varias instancias para ejecutar la aplicación, también aumenta la disponibilidad de esta. Los conjuntos de escalado son un buen punto intermedio entre desarrolladores y personas que toman decisiones comerciales que desean o necesitan compilar aplicaciones en VM, mientras utilizan funciones similares a PaaS para escalar de forma automática y reconfigurar el tráfico del flujo de clientes.

En el capítulo 11, veremos Azure Traffic Manager, que realmente completa estas implementaciones de alta disponibilidad. En este momento, todavía no está preparado para la producción en términos de poder ofrecer múltiples conjuntos de escalado redundantes o instancias de aplicaciones web con tráfico distribuido automáticamente entre ellas. Sin embargo, pronto lo haremos.

9.4 **Laboratorio: Instalación de aplicaciones en el conjunto de escalado o la aplicación web**

Este capítulo ha tenido bastante contenido, así que ahora puede elegir un laboratorio final rápido para los conjuntos de escalado o aplicaciones web. O, si quiere alargar su hora de almuerzo, ¡puede hacer ambos!

9.4.1 **Conjuntos de escalado de máquinas virtuales**

Tiene varias instancias de VM en los conjuntos de escalado, pero en este momento no hacen mucho. Para obtener una descripción general de las diferentes formas de instalar aplicaciones en instancias de VM en un conjunto de escalado, consulte <http://mng.bz/9Ocx>. En la práctica, usaría uno de esos métodos de implementación automáticos; pero por ahora, instale manualmente un servidor web en las instancias de VM como lo hizo en el capítulo 8:

1. ¿Recuerda las reglas NAT del equilibrador de carga? De forma predeterminada, cada instancia de VM de un conjunto de escalado tiene una regla NAT que le permite SSH directamente a ella. Los puertos no están en el puerto TCP 22 estándar. Vea la lista de instancias de VM en un conjunto de escalado y sus números de puertos de la siguiente manera:

```
az vmss list-instance-connection-info \
--resource-group azuremolchapter9 \
--name scalesetmol
```

2. Para conectarse a un puerto específico a través de SSH, utilice el parámetro -p de la siguiente manera (proporcione su propia dirección IP pública y números de puerto):

```
ssh azuremol@40.114.3.147 -p 50003
```

3. Instale un servidor web básico de NGINX en cada instancia de VM con apt install. Piense en cómo lo hizo en el capítulo 8.
4. Para ver el conjunto de escalado en acción, abra la dirección IP pública del equilibrador de carga del conjunto de escalado en un navegador web.

- 5 Si tiene problemas, asegúrese de que el equilibrador de carga ha creado correctamente una regla de equilibrador de carga para el puerto TCP 80 y tiene una sonda de estado asociada para el puerto TCP 80 o su propia sonda de estado HTTP personalizada que busca /health.html en la VM.

9.4.2 Web Apps

Para implementar la aplicación en una aplicación web que ejecuta varias instancias, el proceso es el mismo que para la aplicación web única del capítulo 3. Envíe la aplicación al repositorio de Git local para la aplicación web y, gracias al poder de PaaS, la plataforma Azure implementa ese único código base a varias instancias de aplicaciones web:

- 1 Inicialice un repositorio Git en azure-mol-samples-2nd-ed/09 y luego agregue y confirme los archivos de ejemplo como lo hizo en el capítulo 3:

```
cd azure-mol-samples-2nd-ed/09  
git init && git add . && git commit -m "Pizza"
```

- 2 Su aplicación web tiene un repositorio de Git local. Agregue un control remoto para su aplicación web de la misma manera que lo hizo en el capítulo 3:

```
git remote add webappmolscale <your-git-clone-url>
```

- 3 Inserte este ejemplo en su aplicación web. Esto hace que se confirme un único código, pero la aplicación luego se distribuye a través de varias instancias de la aplicación web:

```
git push webappmolscale master
```

Bases de datos globales con Cosmos DB

Datos. No puede hacerlo sin ellos. Casi todas las aplicaciones que compila y ejecuta crean, procesan o recuperan datos. Tradicionalmente, estos datos se han almacenado en una bases de datos estructurados como MySQL, Microsoft SQL o PostgreSQL. Estas grandes bases de datos estructurados son establecidas y bien conocidas, tienen amplia documentación y tutoriales, y se puede acceder a ellas desde la mayoría de los lenguajes de programación principales.

Gran poder implica gran responsabilidad y, generalmente, estas bases de datos estructurados tradicionales llevan una gran infraestructura y tareas de administración. Eso no quiere decir que no debería usarlas, para nada, pero cuando se trata de aplicaciones que se ejecutan a escala global, tener que crear además clústeres de servidores de bases de datos que replican sus datos y enrutan de forma inteligente a los clientes a su instancia más cercana no es una tarea fácil.

Es ahí donde Azure Cosmos DB se convierte en su mejor amigo. No necesita preocuparse sobre cómo replicar sus datos, garantizar coherencia y distribuir las solicitudes de los clientes. Solo debe agregar los datos a uno de los muchos modelos disponibles y luego elegir dónde desea que estén disponibles sus datos. En este capítulo, aprenderá sobre los modelos de bases de datos no estructurados en Cosmos DB, cómo crear y configurar su base de datos para distribución global y cómo compilar aplicaciones web que utilicen su instancia de Cosmos DB altamente redundante y escalable.

10.1 *¿Qué es Cosmos DB?*

El capítulo 4 comenzó a explorar bases de datos no estructurados con tablas de Azure Storage. El ejemplo era básico, pero los conceptos son el fundamento de Cosmos DB. En primer lugar, demos un paso atrás y analicemos lo que significa una base de datos *estructurada* y *no estructurada*.

10.1.1 Bases de datos estructurados (SQL)

Las bases de datos estructurados son el enfoque más tradicional para el almacenamiento de datos. Una *estructura* o *esquema* de la base de datos define cómo se representan los datos. Los datos se almacenan en tablas; cada fila representa un elemento y un conjunto fijo de valores asignados. Si tomamos el modelo de la pizzería, cada fila de la tabla que guarda los tipos de pizza puede indicar el nombre de la pizza, su tamaño, y el costo. En la figura 10.1 se muestra una base de datos SQL básica.

Base de datos estructurados			
Tabla			
id	pizzaName	tamaño	costo
1	Salchichón	16"	USD 18
2	Vegetariana	16"	USD 15
3	Hawaiana	16"	USD 12

Figura 10.1 En una base de datos estructurados, los datos se almacenan en filas y columnas dentro de una tabla. Cada fila contiene un conjunto fijo de columnas que representan el esquema de la base de datos.

En las bases de datos estructurados, cada servidor debe contener toda la base de datos para que las consultas y la recuperación de información sean correctas. Los datos se juntan en consultas obtenidas de distintas tablas en función de los criterios que el desarrollador compila como parte la consulta estructurada. De ahí proviene el término *Structured Query Language* (SQL) o lenguaje de consulta estructurado. A medida que las bases de datos crecen tanto en tamaño como en complejidad, los servidores que ejecutan la base de datos deben ser lo suficientemente dimensionados para manejar esos datos in-memory. Eso se vuelve difícil y costoso con bases de datos demasiado grandes. Dado que necesitan una estructura, también resulta complicado agregar propiedades y cambiar la estructura más adelante.

10.1.2 Bases de datos (NoSQL) no estructurados

Base de datos no estructurados
<pre>{ "costo": "18", "descripción": "salchichón" } { "costo": "15", "descripción": "vegetariana", "gluten": "libre" } { "costo": "12", "descripción": "hawaiana", "ingredientes": "jamón, piña" }</pre>

Los datos no estructurados en bases de datos NoSQL no se almacenan en tablas de filas y columnas; más bien, se almacenan en matrices dinámicas que le permiten agregar nuevas propiedades para un elemento, según sea necesario. Una gran ventaja de este enfoque es que puede agregar rápidamente un nuevo tipo de pizza o ingrediente especial sin cambiar la estructura subyacente de la base de datos. En una base de datos estructurados, es necesario agregar una nueva columna a una tabla y luego actualizar la aplicación para utilizar la columna adicional. En las bases de datos NoSQL, se agrega otra propiedad a una entrada determinada del código; vea la figura 10.2.

Figura 10.2 En una base de datos no estructurados, los datos se almacenan sin asignaciones fijas de columnas a la fila de una tabla. Puede agregar ingredientes a una sola pizza, por ejemplo, sin actualizar el esquema completo ni otros registros.

Las bases de datos de NoSQL también ofrecen diferentes modelos de base de datos. Estos modelos dan una indicación de cómo se almacenan los datos y se recuperan en la base de datos. El modelo que utilice varía en función del tamaño y el formato de los datos con los que trabaja y de cómo necesite representar los datos de la aplicación. Estos modelos incluyen documento, gráfico y tabla. No se quede entrampado en los modelos por ahora; los diferentes modelos funcionan mejor para diferentes conjuntos de datos no estructurados, dependiendo de cómo necesite relacionar y consultar los datos. La clave está en entender que las bases de datos no estructurados NoSQL tienen un concepto subyacente diferente a la forma en que almacenan y recuperan los datos, lo que puede utilizar a su ventaja a medida que compila y ejecuta aplicaciones en la nube en Azure.

10.1.3 Escalado de bases de datos

¿Recuerda que dijimos que para una base de datos estructurados, normalmente toda la base de datos debe existir en cada servidor? A medida que empieza a tener bases de datos muy grandes, necesita servidores cada vez más grandes para ejecutarlas. Es posible que nunca trabaje con bases de datos que crezcan a cientos de gigabytes o incluso terabytes de tamaño, pero las bases de datos NoSQL sirven para entender cómo las bases de datos crecen y escalan de manera diferente a las bases de datos SQL. La diferencia es que las bases de datos NoSQL suelen escalar de forma horizontal en lugar de vertical.

Hay un límite de cuánto puede escalar verticalmente una VM, es decir, darle más memoria y CPU. Empieza a tener problemas de rendimiento en otras partes de la unidad de proceso al exprimir al máximo el rendimiento de almacenamiento y el ancho de banda de la red. Y eso sin mencionar su billetera (o la billetera de su jefe) cuando ve la cuenta de esas inmensas VM. Como resumen del capítulo 9, la escalabilidad vertical se ilustra en la figura 10.3. Ahora imagine un clúster de esas VM con inmensas bases de datos, porque busca redundancia y resiliencia para su aplicación, ¿verdad?

Por el contrario, la escalabilidad horizontal le permite ejecutar VM de base de datos con menos recursos a un precio más bajo. Para hacer esto, las bases de datos NoSQL dividen los datos en nodos de la base de datos y dirigen las solicitudes desde su aplicación al nodo apropiado. Los demás nodos del clúster no necesitan saber dónde se almacenan todos los datos;

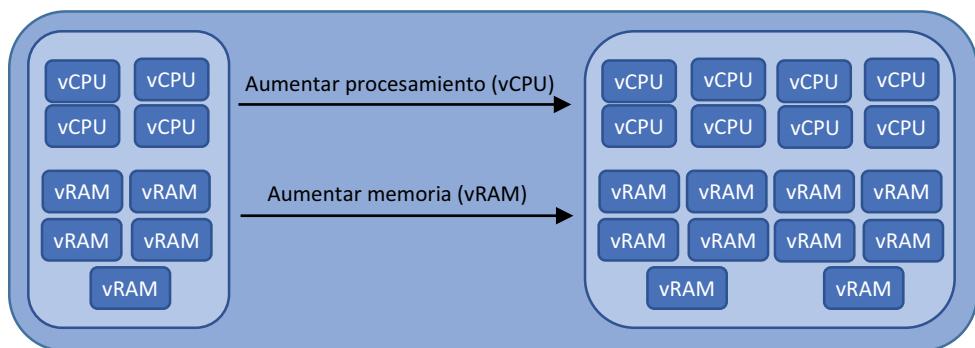


Figura 10.3 Las bases de datos estructurados tradicionales escalan verticalmente. A medida que la base de datos crece, aumenta la cantidad de almacenamiento, memoria y energía de la CPU en el servidor.

solo necesitan responder a sus propias solicitudes. Puede agregar rápidamente nodos a un clúster en respuesta a la demanda del cliente según sea necesario.

Como resultado, en una base de datos NoSQL, no es necesario que toda la base de datos quepa en la memoria de un host. Solo se necesita guardar y procesar una parte de la base de datos, una *partición*. Si su aplicación funciona con grandes cantidades de datos *estructurados*, una base de datos NoSQL puede perjudicar el rendimiento, ya que los diferentes hosts consultan sus datos para luego devolver al cliente. Si tiene una gran cantidad de datos *no estructurados* que procesar, las bases NoSQL pueden optimizar el rendimiento o, al menos, ofrecen la ventaja de administración y eficiencia. En la figura 10.4 se muestra un ejemplo de cómo las bases de datos no estructurados escalan horizontalmente entre hosts.

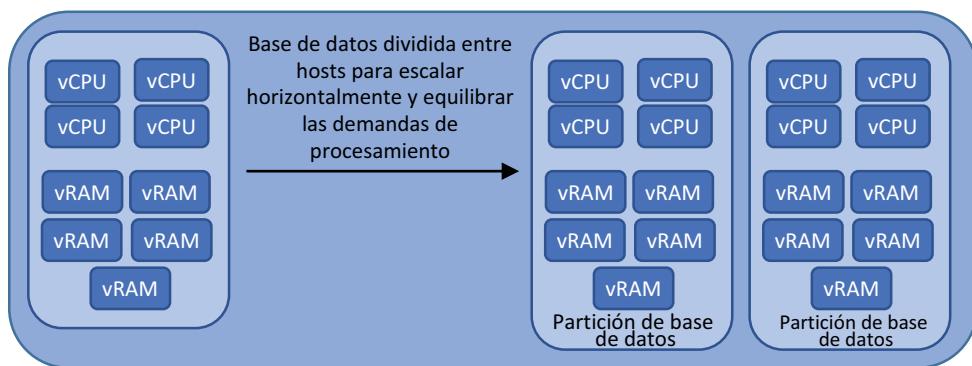


Figura 10.4 Las bases de datos NoSQL no estructurados escalan horizontalmente. A medida que la base de datos crece, se fragmenta en segmentos de datos que se distribuyen en los servidores de base de datos.

10.1.4 Cosmos DB se encarga de todo

Entonces, ¿qué es Cosmos DB? Es una plataforma de bases de datos globalmente distribuida y de escalado automático que le permite utilizar diversas formas de bases de datos NoSQL. Al igual que con servicios como Web Apps, Cosmos DB se encarga de gran parte de la administración que usted debe realizar. Cuando crea una aplicación web, no es necesario configurar equilibradores de carga o clústeres: se eligen las regiones y se puede configurar escalado automático y, a continuación, se carga el código de la aplicación. La plataforma Azure se encarga de cómo replicar y distribuir el tráfico de aplicaciones web de forma altamente disponible. Con Cosmos DB, no debe preocuparse por la gran cantidad de bases de datos que necesita, cuánta memoria asignar o cómo replicar datos para redundancia. Simplemente elige la cantidad de rendimiento que podría necesitar y las regiones para guardar sus datos, y luego empieza a agregar datos.

Este capítulo utiliza un modelo SQL para Cosmos DB, pero los datos se almacenan en formato NoSQL, JSON. Estos pueden ser conceptos nuevos, pero quédese conmigo. Se pueden usar otros modelos, incluido Mongo, Cassandra, Gremlin y Table. La funcionalidad es la misma para todos: escoge el modelo, elige las regiones y agrega los datos. Ese es el poder de Cosmos DB.

10.2 Creación de una cuenta y base de datos de Cosmos DB

Veamos Cosmos DB y las bases de datos no estructuradas en acción, lo que podemos hacer de un par de maneras. La primera es utilizar Azure Portal para crear una cuenta, seleccionar y crear un modelo de base de datos e ingresar datos en la base de datos para que la aplicación pueda consultarla. O bien, puede utilizar la CLI de Azure, Azure PowerShell o kits de desarrollo de software (SDK) específicos del lenguaje para crearlo todo en código. Usemos Azure Portal para que también podamos crear y consultar los datos visualmente.

10.2.1 Creación y relleno de una base de datos de Cosmos DB

En el capítulo 4, creó su primera base de datos NoSQL con una tabla de Azure Storage. Utilicemos Cosmos DB para crear una base de datos similar, esta vez una que ofrezca todas las opciones de redundancia geográfica y replicación para asegurarse de que los clientes puedan pedir pizzas en su tienda en línea sin interrupciones. Vamos a crear una cuenta de Cosmos DB y una base de datos de documentos, y luego agreguemos algunos datos para tres tipos de pizzas, como se muestra en la figura 10.5.

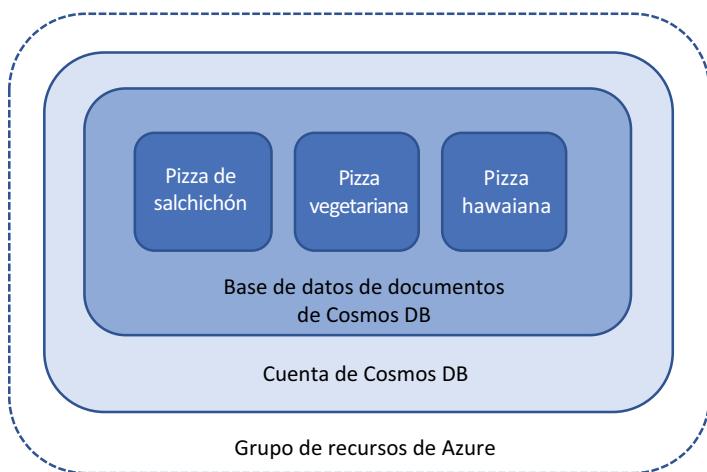


Figura 10.5 En esta sección, creará un grupo de recursos y una cuenta de Cosmos DB. A continuación, creará una base de datos de documentos en esta cuenta y agregará tres entradas para representar un menú básico de su pizzería.

Pruébelo ahora

Para ver Cosmos DB en acción, cree una cuenta a través de Azure Portal:

- 1 Abra Azure Portal y seleccione Crear un recurso en la esquina superior izquierda del panel.
- 2 Busque y seleccione Azure Cosmos DB y, a continuación, elija Crear.
- 3 Elija crear un grupo de recursos, como `azuremolchapter10` y escriba un nombre único para su cuenta de Cosmos DB, como `azuremol`.

- 4 El tipo de modelo que puede utilizar para su base de datos se conoce como API. Para este ejemplo, seleccione Core (SQL) en el menú desplegable.
- 5 Para Ubicación, seleccione Este de EE. UU. Cosmos DB está disponible en todas las regiones de Azure, pero para este ejemplo, la aplicación web que implemente en el laboratorio de fin del capítulo espera que use Este de EE. UU.
- 6 Deje la opción de redundancia geográfica como deshabilitada, junto con cualquier otra característica adicional, como las regiones de multiescritura. En la sección 10.2.2 se profundiza en cómo replicar su base de datos globalmente.

Tráfico seguro con puntos de conexión de servicio

Tiene la opción de conectar su Cosmos DB a una red virtual de Azure con algo llamado *punto de conexión de servicio*. No analizaremos esta opción ahora, pero es una característica interesante que ayuda a asegurar su instancia permitiendo el acceso a la base de datos solo desde una red virtual definida.

Si crea aplicaciones de middleware que utilicen Cosmos DB, o aplicaciones solo internas, puede utilizar un punto de conexión de servicio de red virtual para que el acceso sea desde una red virtual específica, no a través de Internet y con un punto de conexión público. Un número cada vez mayor de servicios de Azure admiten este tipo de puntos de conexión, y es otro ejemplo de cómo ofrecer opciones para asegurar su entorno para que se adapte a los requisitos de su empresa.

- 7 Cuando esté listo, revise y cree su cuenta de Cosmos DB. Crear una cuenta demora unos minutos.

En este momento su base de datos está vacía, así que exploremos cómo puede guardar algunos datos básicos para el menú de su pizzería. Cosmos DB agrupa los datos de una base de datos en algo llamado *contenedor*. No, no es el mismo tipo de contenedor que es la fuerza impulsora de Docker, Kubernetes y las aplicaciones nativas de la nube que puede haber oído. Esta confusión de nombres no es una ventaja, pero sigue conmigo por ahora.

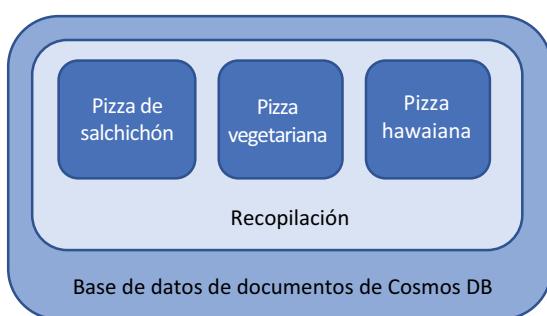


Figura 10.6 Una base de datos de Cosmos DB que utiliza el modelo de documento guarda datos en colecciones. Estas colecciones le permiten agrupar datos para una indexación y consulta más rápidas.

En las bases de datos de Cosmos DB que utilizan el modelo de documento, la información se agrupa lógicamente en contenedores denominados *colecciones*. Otros modelos de API tienen un nombre ligeramente diferente para la entidad de contenedor, como *graph* para la API de Gremlin. Para nuestra API de SQL, las colecciones almacenan datos relacionados que se pueden indexar y consultar rápidamente, como se muestra en la figura 10.6. Las colecciones no son totalmente diferentes de cómo se organiza una base de datos SQL tradicional en tablas, pero ofrecen mucha más flexibilidad a la hora de distribuir los datos para rendimiento o redundancia.

Como Cosmos DB está diseñado para manejar grandes cantidades de datos y para un alto rendimiento, puede elegir cómo dimensionar y controlar el flujo y el costo de esos datos. El rendimiento se calcula en unidades de solicitud por segundo (RU/s), y una unidad de solicitud equivale a 1 KB de datos de documento. Básicamente, usted define el ancho de banda que quiere para su base de datos. En caso de que no lo haya adivinado, cuanto más ancho de banda (RU/s) quiera, más tendrá que pagar. Cosmos DB le muestra la cantidad de datos que está utilizando y el rendimiento que utiliza su aplicación, y normalmente no necesita preocuparse demasiado sobre dimensionar correctamente las cosas. Para su pizzería, iremos de a poco.

Pruébelo ahora

Para crear una colección y llenar algunas entradas de la base de datos, complete los siguientes pasos:

- 1 Busque y seleccione Grupos de recursos en la barra de navegación del lado izquierdo de Azure Portal.
- 2 Seleccione el grupo de recursos donde creó la base de datos de Cosmos DB, como azuremolchapter10.
- 3 Seleccione su cuenta de Cosmos DB en la lista de recursos y, a continuación, elija la página Información general.
- 4 Seleccione para agregar un contenedor.
- 5 Ya que se trata de su primera base de datos, escriba un nombre, como pizzaadb.
- 6 Deje el rendimiento establecido en el valor predeterminado.
- 7 Para el ID de contenedor, ingrese pizzas. Este paso crea un contenedor lógico que puede utilizar para guardar elementos en el menú de la pizzería.
- 8 Escriba una clave de partición de /description para asegurarse de que los tipos de pizza se distribuyan de manera uniforme.

La clave de partición identifica la forma en que se pueden separar los datos en la base de datos. No es realmente necesario en una pequeña base de datos de muestra como esta, pero su uso es una buena práctica a medida que su aplicación se amplía.

- 9 No elija agregar clave única. Las claves definen más lógicamente el contenedor, por ejemplo, para las subdivisiones de alimentos que pueden pedir los clientes. La colección más amplia es para su menú, pero en bases de datos mucho más grandes, podría querer claves de partición para las pizzas, las bebidas, y los postres.
- 10 Para crear la base de datos y la colección, seleccione Aceptar.

Ahora tiene una cuenta de Cosmos DB, una base de datos y una colección, pero Cosmos DB todavía no contiene sus pizzas. Puede importar algunos datos o escribir código que incorpora una gran cantidad de datos. Vamos a crear manualmente tres pizzas para explorar algunas de las herramientas gráficas incorporadas en Azure Portal para buscar, consultar y manipular los datos en su base de datos de Cosmos DB.

Pruébelo ahora

Para agregar algunas entradas a la base de datos, complete los siguientes pasos, como se muestra en la figura 10.7:

- 1 En su cuenta de Cosmos DB, elija Explorador de datos en el menú de la izquierda en la ventana Información general.

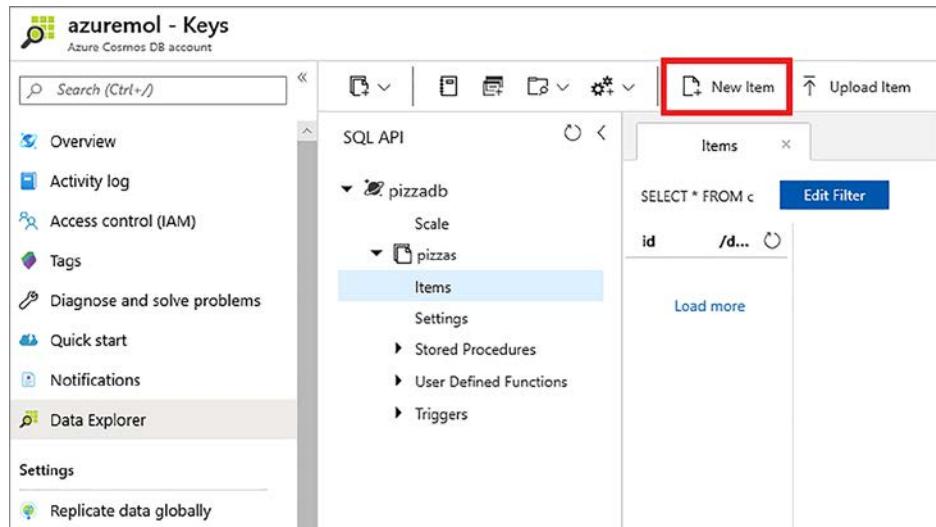


Figura 10.7 Con el Data Explorer en Azure Portal, puede navegar por sus colecciones para consultar o crear nuevos documentos. Esta herramienta gráfica le permite administrar rápidamente su base de datos desde un navegador web.

- 2 Amplíe primero la base de datos pizzadb y luego la colección de pizzas.
- 3 Agregue un nuevo elemento para poner algunas pizzas en la base de datos. Los datos se agregan en formato JSON.
- 4 En el cuadro de texto, sustituya cualquier texto existente por los siguientes datos para crear un nuevo elemento de menú para una pizza básica de pepperoni:

```
{
    "description": "Pepperoni",
    "cost": "18"
}
```

- 5 Para agregar los datos a la base de datos, seleccione Guardar.
- 6 Agregue otra pizza a su menú. Esta vez, agregue una propiedad para indicar que esta pizza tiene masa sin gluten. No necesita hacer nada especial en la base de datos subyacente, simplemente agregue otra propiedad a sus datos. Para agregar otro elemento nuevo, escriba los siguientes datos y seleccione Guardar:

```
{  
    "description": "Veggie",  
    "cost": "15",  
    "gluten": "free"  
}
```

- 7 Agregue un último tipo de pizza. Esta vez, agregue una propiedad que incluya los ingredientes de la pizza. Para agregar otro elemento nuevo, escriba los siguientes datos y seleccione Guardar:

```
{  
    "description": "Hawaiian",  
    "cost": "12",  
    "toppings": "ham, pineapple"  
}
```

Estas tres entradas muestran el poder de una base de datos NoSQL. Agregó propiedades a las entradas sin necesidad de actualizar el esquema de la base de datos. Dos propiedades diferentes mostraron que la pizza vegetariana tiene masa sin gluten y qué ingredientes tiene la pizza hawaiana. Cosmos DB acepta esas propiedades adicionales y los datos ahora están disponibles para sus aplicaciones.

Se agregan algunas propiedades JSON adicionales para cosas como `id`, `_rid` y `_self`. Estas no son propiedades por las que tenga que preocuparse demasiado por ahora. Cosmos DB usa estas propiedades para realizar seguimiento e identificar los datos; no debe editarlos o eliminarlos manualmente.

10.2.2 Agregar redundancia global a una base de datos de Cosmos DB

Ahora tiene una base de datos de Cosmos DB que guarda un menú básico de pizzas en la región Este de EE. UU. ¡Pero su pizzería está lista para abrir franquicias en todo el mundo! Replique los datos de sus pizzas en regiones de Azure en diferentes ubicaciones, cerca de sus nuevos clientes.

¿Para qué sirve hacer esto? Si todos sus clientes leen y escriben datos de la base de datos en una región, es una gran cantidad de posible tráfico cruzando cables bajo el océano y enrutamiento en todo el mundo. Para proporcionar la mejor experiencia de baja latencia a los clientes, puede replicar sus datos a las regiones de Azure disponibles alrededor del mundo, y los clientes podrán conectarse a la réplica más cercana a ellos, como se muestra en la figura 10.8.

Los modelos de coherencia y las garantías se incorporan a la plataforma de Cosmos DB para gestionar automáticamente esta coherencia y replicación de datos. Puede desasignar una o más regiones como ubicación de escritura primaria. Los ejemplos de este libro utilizan un único punto de escritura, pero puede utilizar el soporte de Multimaster para escribir datos en el punto de conexión más cercano para que luego se propague de forma

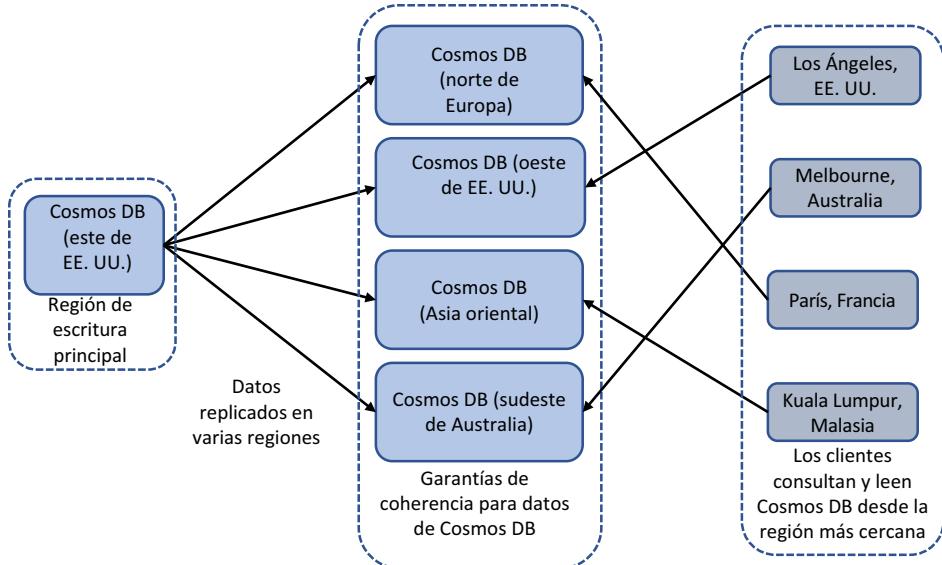


Figura 10.8 Los datos se replican desde una instancia de Cosmos DB primaria a varias regiones de Azure en todo el mundo. Luego, las aplicaciones web se dirigen para que sean leídas desde la región más cercana y los clientes pueden direccionarse dinámicamente a la ubicación más cercana para minimizar la latencia y mejorar los tiempos de respuesta.

asincrónica a otras regiones. Los datos también se replican rápidamente en las regiones leídas que usted designe. Puede controlar el orden de comutación por error para designar regiones de lectura y, con la aplicación, especificar automática o manualmente regiones desde las que se leerán.

Puede definir un modelo de coherencia (que es más una consideración de diseño que operacional) que defina la rapidez con la que se replican las escrituras en varias regiones. Los modelos de coherencia van de *fuerte*, que espera que las escrituras replicadas sean confirmadas por réplicas y así garantizar que las lecturas sean uniformes, a *definitivo*, que es más relajado. El modelo definitivo garantiza que todos los datos se repliquen, pero puede haber un ligero retraso cuando las lecturas de réplicas devuelvan valores diferentes hasta que estén todos sincronizados.

Hay un equilibrio entre una distribución geográfica más limitada, como el modelo de coherencia fuerte y una replicación geográfica más amplia que ofrece el modelo de coherencia definitiva, pero entendiendo que hay un ligero retraso mientras se replican los datos. También hay costos relacionados con el ancho de banda y con el procesamiento, dependiendo de la coherencia y la frecuencia con que desea que se repliquen los datos. La plataforma Azure maneja la replicación subyacente de los datos desde su punto de escritura; no es necesario que compile sus aplicaciones para replicar los datos o determinar la mejor manera de leer los datos desde los puntos de conexión replicados.

En una escala global, esto significa que podría tener varias VM o aplicaciones web como las que creó en capítulos anteriores, pero en diferentes regiones del mundo. Esas aplicaciones se conectan a una instancia de Cosmos DB local para consultar y leer todos sus datos. Mediante algunas útiles funciones de tráfico de red Azure que analizaremos en el capítulo 11, los usuarios pueden ser direccionados automáticamente a una de estas instancias de aplicaciones web locales, que también utilizan una instancia local de Cosmos DB. En caso de interrupciones o mantenimiento regionales, la plataforma completa enruta al cliente a la siguiente instancia más cercana.

En el mundo tradicional de las bases de datos estructurados donde usted administra las VM, la instalación de la bases de datos y la configuración de clúster, se requiere una seria planificación de diseño, ya que toda esta configuración es complicada de implementar. Con Cosmos DB, el proceso requiere apenas tres clics. ¡En serio!

Pruébelo ahora

Complete los siguientes pasos para replicar sus datos de Cosmos DB globalmente:

- 1 Busque y seleccione Grupos de recursos en la barra de navegación del lado izquierdo de Azure Portal.
- 2 Seleccione el grupo de recursos donde creó la base de datos de Cosmos DB, como `azuremolchapter10`.
- 3 Seleccione su cuenta de Cosmos DB en la lista de recursos. Esos dos clics fueron gratis, pero ¡empiece a contar desde aquí!
- 4 Seleccione la opción de menú a la izquierda para replicar los datos de manera global. En el mapa, que muestra todas las regiones de Azure disponibles, se muestra que su base de datos está actualmente disponible en la región Este de EE. UU. (Figura 10.9).

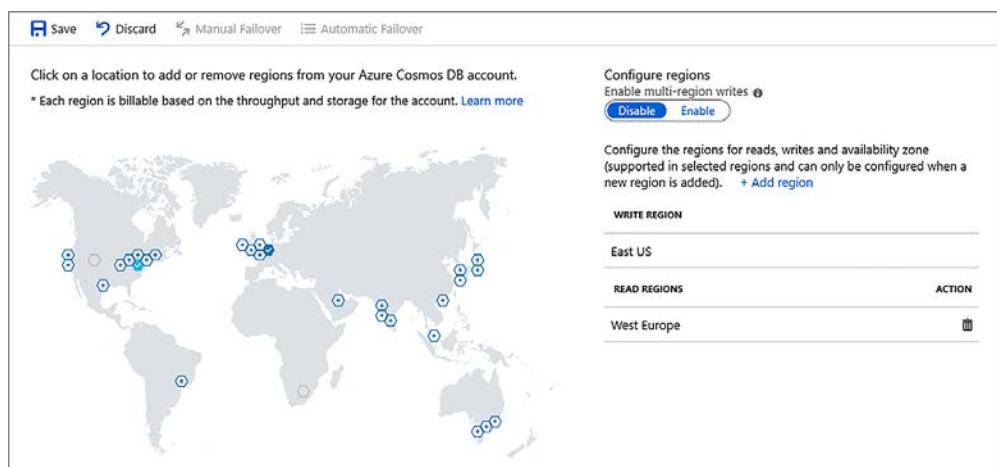


Figura 10.9 Seleccione una región de Azure de cualquier parte del mundo para replicar su base de datos de Cosmos DB y, a continuación, elija Guardar. Estos son todos los pasos requeridos para distribuir globalmente sus datos.

- 5 Elija Oeste de Europa y, a continuación, seleccione Guardar. Puede elegir cualquier región Azure que desee, pero para el laboratorio de fin del capítulo sus datos se deben replicar a Oeste de Europa. Replicar los datos en la región que seleccionó y que luego queden disponibles en línea para que sus aplicaciones los utilicen puede tomar un rato.

¡Vaya contando esos clics! Tres clics, ¿verdad? Seamos generosos y consideremos los dos primeros clics para seleccionar el grupo de recursos y la cuenta Cosmos DB. Por lo tanto, en no más de cinco clics y en cuestión de segundos, creó una instancia de réplica de su base de datos que permite a sus aplicaciones acceder a los datos desde la región más cercana. ¿Puede hacerlo con un clúster MySQL tradicional? Por favor, envíeme un tweet a @fouldsy si puede hacerlo rápidamente fuera de Cosmos DB.

Con su base de datos ahora distribuida globalmente, ¿necesita hacer varios cambios en el código para determinar a qué región de Cosmos DB se conectará? ¿Cómo puede mantener todas estas diferentes versiones de sus aplicaciones en la región Azure donde se ejecutan? Fácil, ¡deje que la plataforma Azure determine todo!

10.3 **Acceso a los datos distribuidos globalmente**

En su gran mayoría, la plataforma Azure determina la mejor ubicación para hablarle a su aplicación. Una aplicación normalmente necesita leer y escribir datos. Puede definir las directivas de comutación por error para su base de datos de Cosmos DB que controla la ubicación de escritura principal. Esta ubicación de escritura actúa como el nodo central para garantizar que los datos se repliquen uniformemente en todas las regiones. Pero su aplicación web puede normalmente leer de varias regiones disponibles para acelerar las consultas y devolver datos al cliente. Todo esto es manejado por llamadas REST.

Veamos lo que sucede con la CLI de Azure cuando pide información sobre una base de datos de Cosmos DB. Este proceso es como una aplicación que hace una conexión a una base de datos, pero impide que se meta demasiado profundo en el código.

Pruébelo ahora

Utilice az cosmosdb show para encontrar información sobre su ubicación de lectura y escritura, como se muestra a continuación:

- 1 Abra Azure Portal en un navegador web y, a continuación, abra Cloud Shell.
- 2 Utilice az cosmosdb show para ver las ubicaciones de lectura y escritura de su base de datos de Cosmos DB.

Escriba el nombre del grupo de recursos y el nombre de la base de datos que creó en los ejercicios anteriores "Pruébelo ahora". En el ejemplo siguiente, el grupo de recursos azuremolchapter10 y el nombre de la base de datos de Cosmos DB es azuremol:

```
az cosmosdb show \
--resource-group azuremolchapter10 \
--name azuremol
```

Este comando devuelve una gran cantidad de resultados, así que examinemos las dos partes clave: leer ubicaciones y escribir ubicaciones. Este es un ejemplo del resultado de la sección `readLocations`:

```
"readLocations": [
  {
    "documentEndpoint": "https://azuremol-eastus.documents.azure.com:443/",
    "failoverPriority": 0,
    "id": "azuremol-eastus",
    "isZoneRedundant": "false",
    "locationName": "East US",
    "provisioningState": "Succeeded"
  },
  {
    "documentEndpoint": 
      "https://azuremol-westeuropa.documents.azure.com:443/",
    "failoverPriority": 1,
    "id": "azuremol-westeuropa",
    "isZoneRedundant": "false",
    "locationName": "West Europe",
    "provisioningState": "Succeeded"
  }
],
```

Cuando la aplicación hace una conexión a una base de datos de Cosmos DB, puede especificar una directiva de conexión. Si las bases de datos no son normalmente lo suyo, piense en una conexión Open Database Connectivity (ODBC) básica que puede crear en una máquina Windows. Normalmente, la cadena de conexión define un nombre de host, un nombre de base de datos, un puerto y credenciales. Cosmos DB no es diferente. Puede conectarse a Cosmos DB desde varios lenguajes, incluyendo .NET, Python, Node.js y Java. Los lenguajes pueden diferir, pero todos los SDK tienen un valor similar: descubrimiento del punto de conexión. La directiva de conexión tiene dos propiedades principales importantes:

- *Descubrimiento automático del punto de conexión*: el SDK lee todos los puntos de conexión disponibles de Cosmos DB y utiliza el orden de conmutación por error especificado. Este enfoque asegura que su aplicación siempre siga el orden que especifique en la base de datos. Por ejemplo, es posible que desee que todas las lecturas pasen por el Este de EE. UU. y solo utilizar Oeste de Europa cuando haya mantenimiento en la ubicación principal.
- *Ubicaciones del punto de conexión preferido*: usted especifica las ubicaciones que desea utilizar. Un ejemplo es si implementa su aplicación en Oeste de Europa y quiere asegurarse de que utiliza el punto de conexión de Oeste de Europa. Se va perdiendo un poco de flexibilidad a medida que se agregan o eliminan los puntos de conexión, pero se asegura de que el punto de conexión predeterminado esté cerca de la aplicación sin necesidad de un enruteamiento de red más avanzado para determinar esto.

Por lo general, la aplicación permite que el SDK de Cosmos DB se encargue de esta tarea. La aplicación no cambia la forma de manejar la conexión a la base de datos: solo sabe que *puede* conectarse a diferentes ubicaciones. Pero el SDK es lo que *establece* la conexión y utiliza este reconocimiento de la ubicación.

En la figura 10.10 se muestra un enfoque simplificado de cómo se utiliza este reconocimiento de ubicación entre la aplicación y el SDK. Una vez más, el lenguaje no importa, y el enfoque es el mismo: la figura usa el SDK de Python porque en ese lenguaje se han escrito un par de ejemplos. En este ejemplo, también se supone que está utilizando ubicaciones de punto de conexión automático.

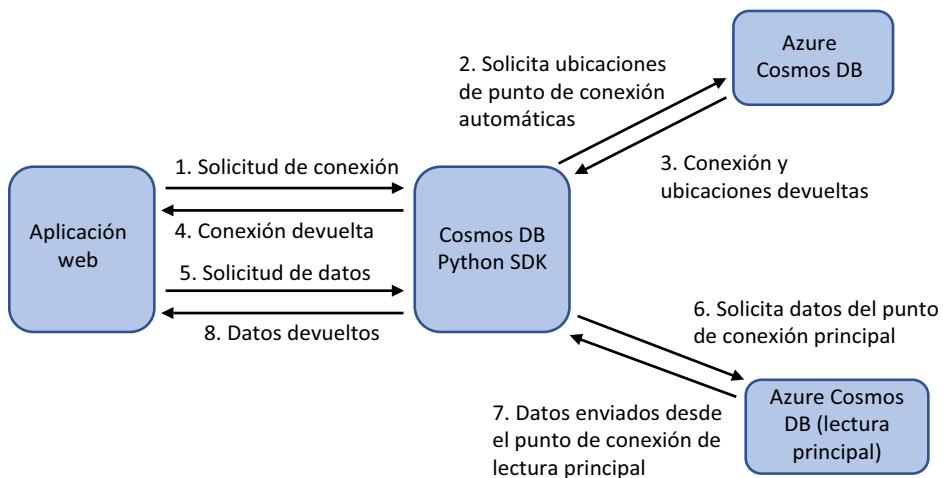


Figura 10.10 Flujo de solicitudes a través de un SDK de Cosmos DB cuando una aplicación utiliza reconocimiento de la ubicación para consultar Cosmos DB.

Los pasos ilustrados en la figura 10.10 son los siguientes:

1. Su aplicación necesita establecer una conexión a una base de datos de Cosmos DB. En la directiva de conexión, habilita el descubrimiento automático de puntos de conexión. La aplicación utiliza el SDK de Cosmos DB para establecer una conexión con la base de datos.
2. El SDK de Cosmos DB hace una solicitud de conexión e indica que desea utilizar ubicaciones automáticas de los puntos de conexión.
3. Se devuelve una conexión basándose en las credenciales y la base de datos solicitadas.
4. El SDK devuelve un objeto de conexión para que utilice la aplicación. La información de ubicación se obtiene de la aplicación.
5. La aplicación solicita algunos datos de la base de datos de Cosmos DB. El SDK se utiliza de nuevo para consultar y obtener los datos.
6. El SDK utiliza la lista de puntos de conexión disponibles y hace la solicitud al primer punto de conexión disponible. A continuación, el SDK utiliza el punto de conexión para consultar los datos. Si el punto de conexión principal no está disponible, como durante mantenimiento, se utiliza automáticamente la siguiente ubicación del punto de conexión.

- 7 Cosmos DB devuelve los datos de la ubicación del punto de conexión.
- 8 El SDK pasa los datos de Cosmos DB de vuelta a la aplicación para analizarlos y mostrarlos según sea necesario.

Lo último que hay que ver en Cosmos DB son las claves de acceso, que permiten controlar quién puede acceder a los datos y qué permisos tiene. Las claves se pueden volver a generar, y al igual que con las contraseñas, puede implementar una directiva para realizar periódicamente este proceso de regeneración de claves. Para acceder a los datos distribuidos en Cosmos DB, necesita obtener sus claves. Azure Portal ofrece la forma de ver todas las claves y cadenas de conexión de su base de datos.

Pruébelo ahora

Complete los siguientes pasos para ver las claves de su cuenta Cosmos DB:

- 1 Busque y seleccione Grupos de recursos en la barra de navegación del lado izquierdo de Azure Portal.
- 2 Seleccione el grupo de recursos donde creó la base de datos de Cosmos DB, como `azuremolchapter10`.
- 3 Seleccione su cuenta de Cosmos DB en la lista de recursos.
- 4 En el lado izquierdo, seleccione Claves.
- 5 Anote el identificador URI y la clave principal (figura 10.11). Utilizará estos valores en el laboratorio de fin del capítulo.

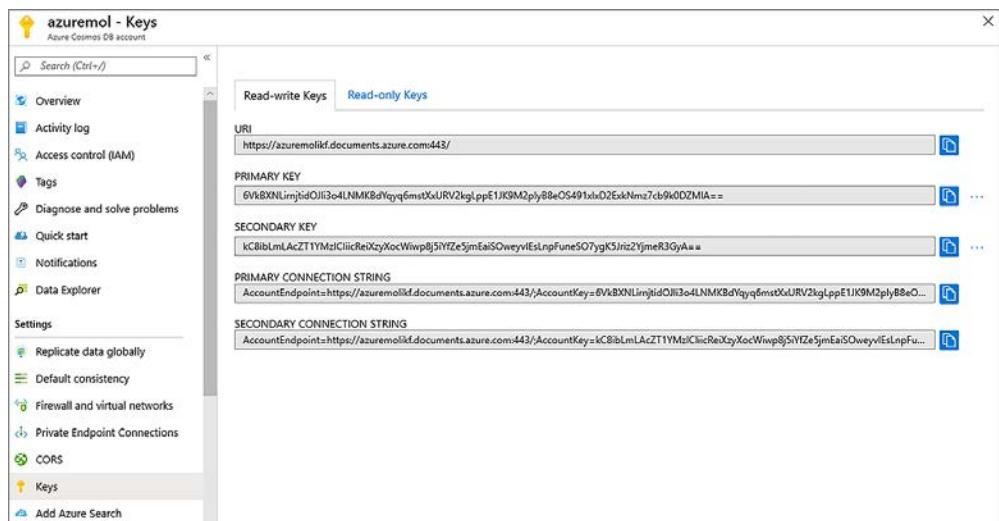


Figura 10.11 La sección Claves de su cuenta Cosmos DB muestra la información de conexión y las claves de acceso. Necesita esta información al compilar y ejecutar aplicaciones, como en el laboratorio de fin del capítulo.

En Cosmos DB se realizan muchas acciones para distribuir sus datos y permitir que sus aplicaciones lean y escriban desde las ubicaciones más apropiadas. Pero esa es precisamente la idea. El conocimiento de lo que hace el servicio Cosmos DB le ayuda a diseñar y planificar su aplicación, o a solucionar problemas si las aplicaciones no permiten que el SDK realice las operaciones de lectura y escritura necesarias. Sin embargo, no es necesario preocuparse sobre cómo y cuándo se realizan esas acciones; simplemente céntrese en sus aplicaciones y utilice los servicios Azure como Cosmos DB para proporcionar la funcionalidad y los beneficios de la nube que le permiten operar a escala global.

10.4 *Laboratorio: Implementación de una aplicación web que usa Cosmos DB*

En la sección 10.2.2, distribuyó globalmente su base de datos de Cosmos DB. Luego, repasamos un montón de teoría sobre cómo las aplicaciones web pueden leer desde ubicaciones de todo el mundo. Ahora probablemente quiera ver a Cosmos DB en acción, ¡así que esta es su oportunidad! En este laboratorio, se utiliza la aplicación web básica de los capítulos anteriores, pero esta vez, el menú de la pizza proviene de los elementos que agregó a la base de datos Cosmos DB en un ejercicio anterior de "Pruébelo ahora":

- 1 Cree una aplicación web en Azure Portal.
- 2 Como la pizzería ya no es una página HTML básica, elija Node LTS para el tiempo de ejecución que se ejecuta en Linux.
- 3 Cuando la aplicación web esté lista, cree un origen de implementación (repositorio Git local). Los pasos son los mismos que cuando creó uno en capítulos anteriores, como el capítulo 3, así que revise esos ejercicios si necesita refrescar la memoria.
- 4 Abra Cloud Shell. En capítulos anteriores, obtuvo una copia de los ejemplos de Azure de GitHub. Si no lo hizo, consiga una copia de la siguiente manera:

```
git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
```

- 5 Cámbielo al directorio que contiene el ejemplo de aplicación web de Cosmos DB.

```
cd ~/azure-mol-samples-2nd-ed/10/cosmosdbwebapp
```
- 6 Edite el archivo de configuración con la URL de base de datos y la clave de acceso que copió en el ejercicio anterior "Pruébelo ahora" para ver sus claves de Cosmos DB:

```
nano config.js
```

- 7 Escriba el archivo presionando Ctrl-O, y luego salga presionando Ctrl-X.
- 8 Agregue y confirme sus cambios en Git con el siguiente comando:

```
git init && git add . && git commit -m "Pizza"
```

- 9 Cree un vínculo al nuevo repositorio de Git en el espacio de ensayo con `git remote add azure`, seguido de su URL de implementación de Git.
- 10 Utilice `git push azure master` para insertar los cambios a su aplicación web.
- 11 Seleccione la dirección URL de la aplicación web desde la ventana Información general de Azure Portal.
- 12 Abra esta URL en un navegador web para ver su pizzería, que ahora está basada en Cosmos DB, como se muestra en la figura 10.12.

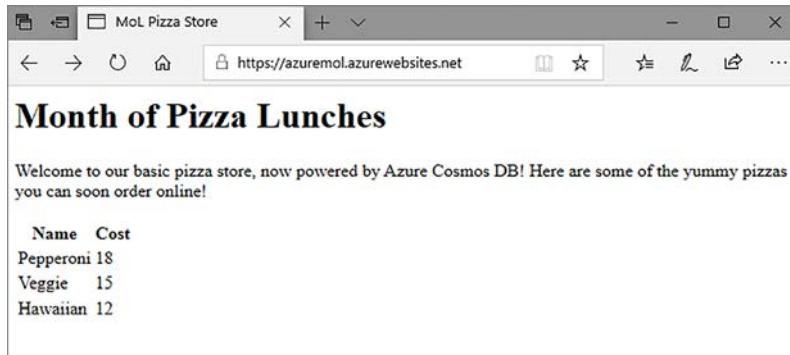


Figura 10.12 La aplicación web básica de Azure muestra su breve menú de pizzas en función de los datos de la base de datos de Cosmos DB. Se muestra la pizzería de los capítulos anteriores, pero ahora Cosmos DB entrega la lista de pizzas y sus precios. El sitio sigue siendo básico, ya que el objetivo es que vea el servicio en acción y comprenda cómo podría empezar a crear sus propias aplicaciones.

Administración del tráfico de redes y enrutamiento

La resolución del Sistema de nombres de dominio (DNS) está al centro de prácticamente todas las conexiones digitales que realiza. Es la forma de navegar por la web, recibir correos electrónicos, ver Netflix, y hacer llamadas por Skype. DNS es el mecanismo que traduce un nombre, como manning.com, a una dirección IP. Cuando quiero aprender un nuevo tema, no necesito recordar 35.166.24.88; simplemente ingreso manning.com en un navegador web y busco los libros que quiero. Los dispositivos de red enrutan el tráfico basado en direcciones IP, así que necesita un enfoque que ayude a aquellos que tenemos mala memoria a hacer cosas como comprar un libro o pedir una pizza en línea.

En los últimos capítulos, hemos dedicado bastante tiempo a aprender cómo compilar aplicaciones que pueden escalar, están altamente disponibles y se distribuyen globalmente. Una de las últimas piezas que falta es cómo dirigir a los clientes de todo el mundo a la instancia de aplicación más adecuada, normalmente la instancia más cercana a ellos. Azure Traffic Manager facilita el enrutamiento automático de los clientes a las instancias de su aplicación basándose en el rendimiento o la ubicación geográfica. En este capítulo, analizaremos cómo puede crear y administrar zonas DNS en Azure y luego cómo utilizar Traffic Manager para enrutar a los clientes con consultas DNS, como se muestra en la figura 11.1.

11.1 ¿Qué es Azure DNS?

No necesita una comprensión profunda de cómo funciona DNS para completar este capítulo y usar Azure DNS. La figura 11.2 muestra una descripción general de alto nivel sobre cómo un usuario consulta un servicio DNS para obtener la dirección IP de una aplicación web. Pueden ocurrir una gran cantidad de pasos secundarios alrededor de los pasos 1 y 2, así que si le sobra un poco de tiempo a la hora de almuerzo cuando finalice este capítulo, tómese un tiempo y lea sobre cómo funcionan las consultas DNS y la recursividad.

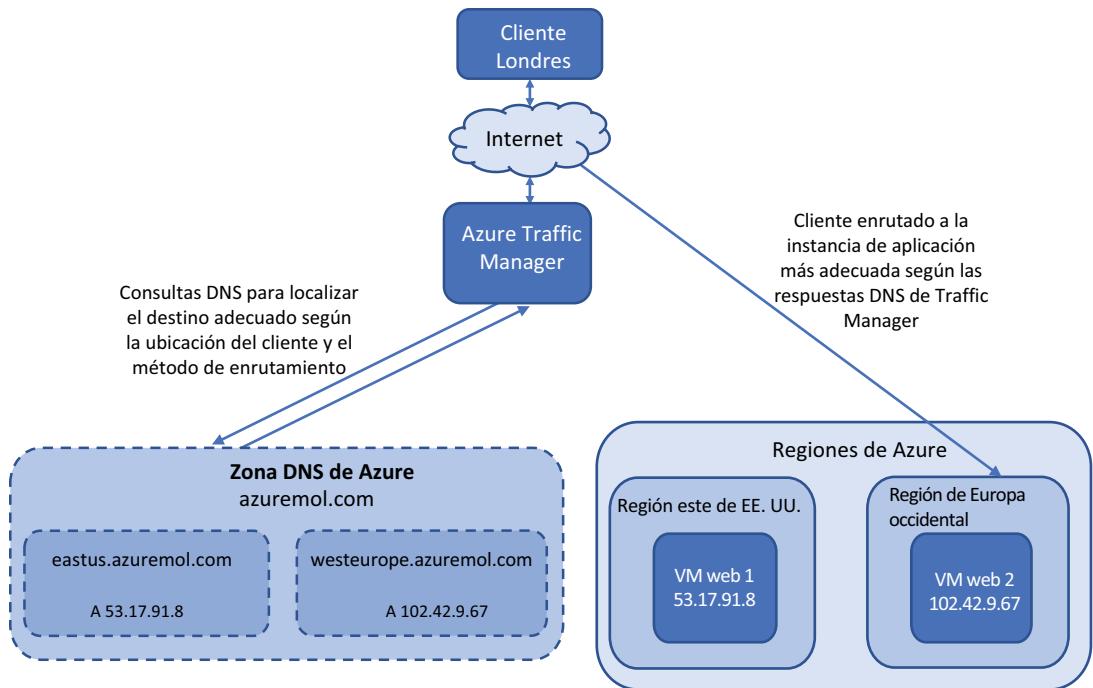


Figura 11.1 En este capítulo, examinaremos cómo puede crear zonas DNS en Azure DNS. Para minimizar la latencia y mejorar los tiempos de respuesta, se puede utilizar Traffic Manager para consultar DNS y dirigir a los clientes a su instancia de aplicación más cercana.

Azure DNS funciona de la misma manera que cualquier solución DNS existente que pueda utilizar o conocer. Su zona y registros se almacenan en Azure, y los servidores de nombres que responden a las consultas DNS se distribuyen globalmente a través de los centros de datos de Azure.

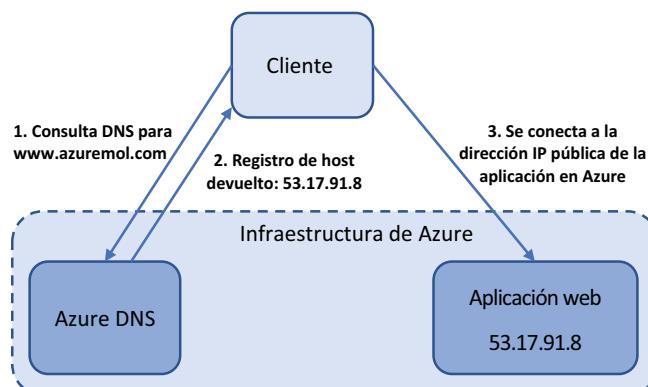


Figura 11.2 Este flujo simplificado de tráfico DNS muestra cómo un usuario envía una solicitud DNS para `www.azuremol.com` a un servidor DNS, recibe una respuesta que contiene la dirección IP asociada, y luego se puede conectar a la aplicación web.

Azure DNS es compatible con todos los tipos de registros que esperaría en un servicio DNS normal. Se pueden crear tanto registros IPv4 como IPv6. Los tipos de registro son los siguientes:

- *A*: registros de host IPv4, para dirigir a los clientes a sus aplicaciones y servicios.
- *AAAA*: registros de host IPv6, para los más entusiastas que utilizan IPv6 para dirigir a los clientes a sus aplicaciones y servicios.
- *CNAME*: nombre canónico, o alias, registros, como para proporcionar un nombre corto que sea más fácil de usar que el nombre de host completo de un servidor.
- *MX*: registros de intercambio de correo para distribuir el tráfico de correo electrónico a sus servidores o proveedor de correo.
- *NS*: registros de servidor de nombres, que incluyen registros generados automáticamente para los servidores de nombres Azure.
- *PTR*: registros de puntero, para consultas de DNS inversas para asignar direcciones IP a nombres de host.
- *SOA*: registros de inicio de autoridad, que incluyen registros generados automáticamente para los servidores de nombres Azure.
- *SRV*: registros de servicio, para proporcionar descubrimiento de servicios de red, por ejemplo, como para identidad.
- *TXT*: registros de texto, como por ejemplo, para el Marco de protección del remitente (SPF) o DomainKeys Identified Mail (DKIM).

En una configuración típica de DNS, usted configura varios servidores DNS. Incluso con la distribución geográfica de esos servidores para redundancia, los clientes pueden consultar un servidor de nombres al otro lado del mundo. Esos milisegundos necesarios para consultar, resolver y luego solicitar una respuesta para la aplicación web pueden sumarse cuando hay muchos clientes que quieren pedir una pizza.

Una zona DNS Azure se replica globalmente a través de los centros de datos de Azure. La red *Anycast* garantiza que cuando un cliente realiza una consulta DNS a su dominio, el servidor de nombres disponible más cercano responde a su solicitud. ¿Cómo funciona el enrutamiento Anycast? Normalmente, se anuncia una sola dirección IP en varias regiones. En lugar de usar una simple consulta DNS que resuelve de vuelta a una única dirección IP que solo existe en una ubicación, el enrutamiento Anycast permite que la infraestructura de red determine de forma inteligente de dónde proviene la solicitud, y dirige al cliente a la región publicada más cercana. Este enrutamiento permite a sus clientes conectarse a la aplicación web con mayor rapidez y le proporciona una mejor experiencia general.

No necesita ser un experto en redes para entender completamente cómo funciona esto, ¡Azure lo hace automáticamente! Al combinar Azure DNS con Azure Traffic Manager (sección 11.2), no solo devuelve las consultas DNS de los servidores de nombres más cercanos, sino que también conecta a los clientes con la instancia de aplicación más cercana. ¡Haga que esos milisegundos cuenten!

11.2 Delegación de un dominio real a Azure DNS

Cuando registra un dominio real, su proveedor le proporciona una interfaz de administración y herramientas para administrar ese dominio. Para permitir a los clientes acceder a sus servicios y utilizar la zona y los registros de Azure DNS, delegue la

autoridad de su dominio a los servidores de nombres Azure. Esta delegación hace que todas las consultas DNS se dirijan inmediatamente a los servidores de nombres Azure, como se muestra en la figura 11.3. Actualmente, Azure no le permite comprar y registrar dominios dentro de la plataforma, por lo que necesita comprar el nombre de dominio a través de un registrador externo y luego dirigir los registros NS a los servidores de nombres Azure.

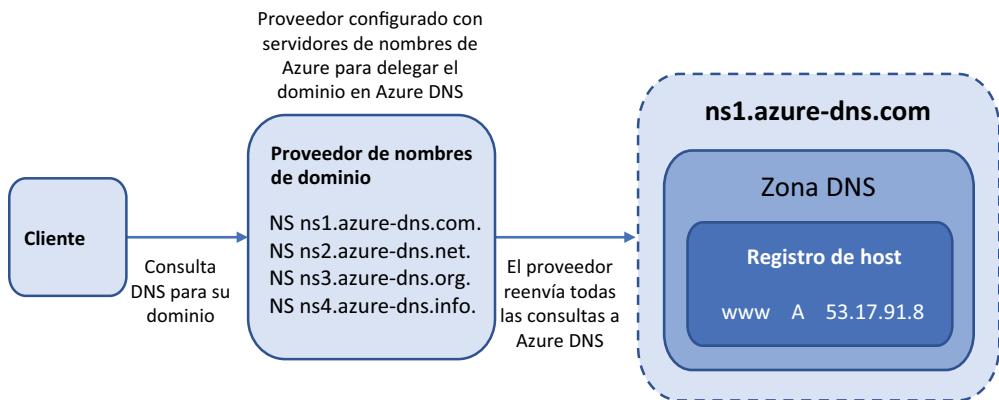


Figura 11.3 Para delegar su dominio a Azure, configure su proveedor de dominio actual con las direcciones de servidor de nombres Azure. Cuando un cliente realiza una consulta DNS para su dominio, las solicitudes se envían directamente a los servidores de nombres Azure de su zona.

¿Por qué delegar su DNS a Azure? Para simplificar la administración y las operaciones. Si crea servicios adicionales, ajusta la configuración del equilibrador de carga o desea mejorar los tiempos de respuesta con DNS replicado globalmente, Azure proporciona esa única interfaz de administración para completar esas tareas. Cuando sus zonas DNS se hospedan en Azure, también puede implementar algunas de las funciones de seguridad de Resource Manager que se analizan en el capítulo 6: funciones como control de acceso basado en roles (RBAC) para limitar y auditar el acceso a las zonas DNS y los bloqueos de recursos para evitar la eliminación accidental, o incluso maliciosa, de la zona.

La mayoría de los registradores de dominio proporcionan interfaces y controles más bien básicos para administrar zonas y registros DNS. Para reducir los gastos generales de administración y mejorar la seguridad, Azure DNS le permite utilizar la CLI de Azure, Azure PowerShell o API REST para agregar o editar registros. Los equipos de operaciones pueden utilizar las mismas herramientas y flujos de trabajo para incorporar nuevos servicios; y si se presentan problemas, a menudo es más fácil solucionarlos cuando puede comprobar que DNS funciona según lo esperado, sin introducir la variable de un proveedor DNS externo.

Por lo tanto, si está convencido de que existe una lógica para delegar su dominio a Azure DNS, ¿a qué servidores de nombres Azure dirige su dominio? Si crea una zona DNS Azure, los servidores de nombres se enumeran en el portal, como se muestra en la figura 11.4. También puede acceder a estas direcciones de servidor de nombres con la CLI de Azure o Azure PowerShell.

No ha habido ejercicios "Pruébelo ahora" en las páginas anteriores, porque a menos que compre y configure un dominio real, no puede probar cómo enrutar tráfico

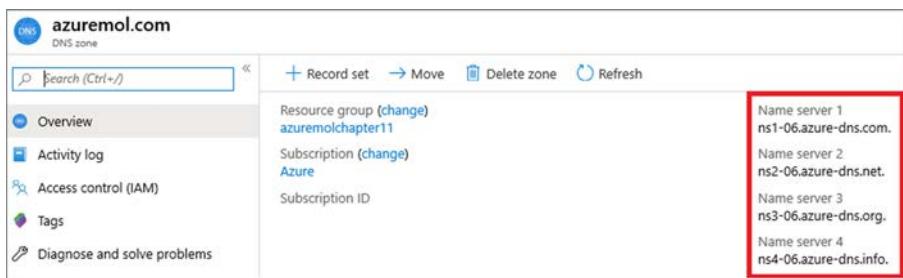


Figura 11.4 Puede ver los servidores de nombres Azure para su zona DNS en Azure Portal, la CLI de Azure o Azure PowerShell.

real. Puede crear una zona DNS Azure sin un dominio real, pero no se puede enrutar ningún tráfico. En la vida real, actualiza los registros NS con su proveedor actual para dirigir las consultas de su dominio a los servidores de nombres Azure. Puede tomar de 24 a 48 horas (aunque normalmente es mucho menos tiempo) para que la delegación de su dominio se propague en la jerarquía global de DNS, este comportamiento puede causar breves interrupciones para los clientes que accedan a su aplicación.

11.3 Enrutamiento global y resolución con Traffic Manager

En capítulos anteriores, aprendió acerca de las aplicaciones altamente disponibles que se distribuyen globalmente. El objetivo final es varias instancias de aplicaciones web o VM en diferentes regiones o continentes, que se conectan a una instancia de Cosmos DB cercana. Pero, ¿cómo hace que sus clientes se conecten a la VM o aplicación web más cercana que ejecuta su aplicación?

Azure Traffic Manager es un servicio de red que actúa como un destino central para sus clientes. Usemos el ejemplo de una aplicación web en la dirección www.azuremol.com. En la figura 11.5 se proporciona una visión general de cómo Traffic Manager enruta a los usuarios a la aplicación disponible más cercana.

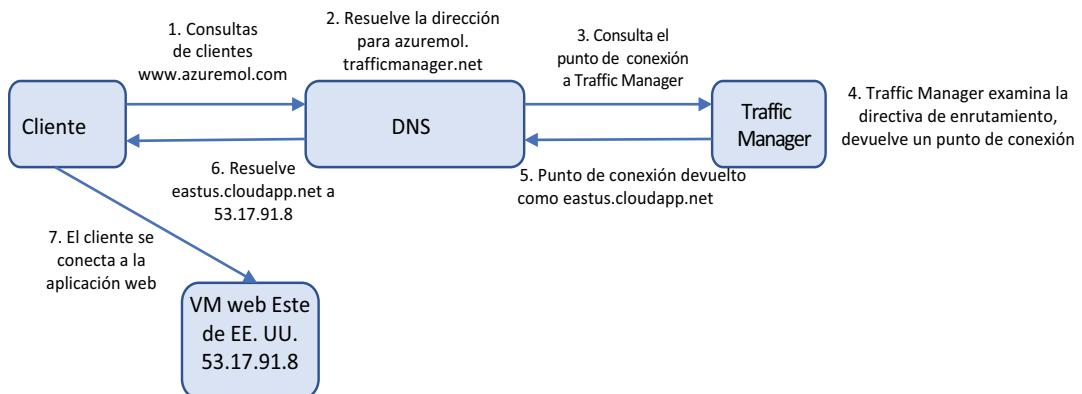


Figura 11.5 El cliente envía una consulta DNS a un servicio DNS para www.azuremol.com. El servicio DNS reenvía la consulta a Traffic Manager, que devuelve un punto de conexión basándose en el método de enruteamiento que se utiliza. El punto de conexión se devuelve a una dirección IP que el cliente utiliza para conectarse a la aplicación web.

Traffic Manager no realiza la función de un equilibrador de carga del que aprendió en el capítulo 8. Como muestra la figura 11.5, Traffic Manager enruta el tráfico a una IP pública. Examinemos el flujo de tráfico un poco más de cerca:

- 1 El usuario realiza una consulta DNS para www.azuremol.com. Su servidor DNS contacta a los servidores de nombres para azuremol.com (que podrían ser los servidores de nombres Azure si usa Azure DNS) y solicita el registro para www.
- 2 El host www determina un registro CNAME que dirige a azuremol.trafficmanager.net.
- 3 El servicio DNS remite la solicitud DNS a los servidores de nombres Azure para trafficmanager.net.
- 4 A continuación, Traffic Manager examina la solicitud y determina un punto de conexión al cual dirigir al usuario. Se examina el estado y funcionamiento del punto de conexión, así como con los equilibradores de carga Azure. También se revisa el método de enrutamiento de Traffic Manager. Los métodos de enrutamiento que Traffic Manager puede utilizar son los siguientes:
 - *Prioridad*: permite controlar el orden en el que se accede a los puntos de conexión.
 - *Ponderado*: distribuye el tráfico a través de los puntos de conexión basándose en una métrica de ponderación asignada.
 - *Rendimiento*: enrutamiento basado en la latencia de los usuarios a un punto de conexión final para que el usuario reciba el tiempo de respuesta más rápido posible.
 - *Geográfico*: asocia puntos de conexión con una región geográfica y dirige a los usuarios basándose en su ubicación.
- 5 Traffic Manager devuelve el punto de conexión eastus.cloudapp.net al servicio DNS.
- 6 El servicio DNS busca el registro DNS para eastus.cloudapp.net y devuelve el resultado de la consulta al cliente.
- 7 Con la dirección IP del punto de conexión solicitado, el cliente se contacta directamente con la aplicación web. En este punto, el tráfico podría llegar a la dirección IP pública de un equilibrador de carga Azure en lugar de llegar directamente a la VM.

Como puede ver, la función de Traffic Manager es determinar un punto de conexión de aplicación determinado para dirigir a los clientes. Algunas comprobaciones de estado que monitorean el estado de los puntos de conexión, similar a los sondeos de estado del equilibrador de carga que aprendió en el capítulo 8. Además, puede definir una prioridad o un mecanismo de enrutamiento de tráfico ponderado para distribuir a los usuarios a través de un conjunto de puntos de conexión disponibles, similar también al equilibrador de carga. Normalmente, Traffic Manager dirige el tráfico a un equilibrador de carga Azure o Application Gateway o a una implementación de Web Apps.

Azure Front Door

Traffic Manager, que veremos en esta sección, es ideal para distribuir y enrutar el tráfico globalmente. Funciona con cualquier tipo de punto final de Internet, no solo con recursos en Azure. El enrutamiento de tráfico está basado en DNS y no examina la aplicación real en sí.

Si necesita una distribución del tráfico a nivel de aplicación y la capacidad de realizar una descarga TLS/SSL o un enrutamiento de solicitudes HTTP/HTTPS, la puerta de entrada de Azure lo ayuda. Traffic Manager y Front Door ofrecen el mismo tipo de servicio

(continuación)

y opciones de configuración, pero Front Door está diseñado específicamente para que funcione en el nivel de aplicación. Front Door también tiene algunos trucos de rendimiento fantásticos, como el TCP dividido para interrumpir las conexiones en piezas más pequeñas y reducir la latencia.

En el capítulo 8, analizamos los equilibradores de carga y Application Gateway mencionada, que funciona en el nivel de la aplicación y hace cosas como la descarga de TLS. El enfoque en el capítulo estaba en los equilibradores de carga para ayudarle a aprender los conceptos básicos, con qué Application Gateway se compilaría. Lo mismo ocurre aquí. Nos centramos en Traffic Manager en este capítulo, aunque muchos de los mismos conceptos y opciones de configuración, como las opciones de enrutamiento, también están disponibles para Azure Front Door. Como sucede con la mayoría de las cosas en Azure, lo que se debe utilizar en cada servicio depende de las aplicaciones que se ejecutan y de sus necesidades.

11.3.1 Creación de perfiles de Traffic Manager

Traffic Manager utiliza perfiles para determinar qué método de enrutamiento utilizar y cuáles son los puntos de conexión asociados para una solicitud dada. Para continuar con el tema de los capítulos anteriores sobre una aplicación distribuida globalmente, sus usuarios deben utilizar la aplicación web más cercana a ellos. Si vuelve a observar los métodos de enrutamiento, tiene dos formas de hacerlo:

- *Enrutamiento de rendimiento*: el cliente se enruta al punto de conexión con la latencia más baja, en relación con el origen de la solicitud. Este método de enrutamiento proporciona cierta inteligencia y siempre permite que Traffic Manager reenvíe al cliente a un punto de conexión disponible.
- *Enrutamiento geográfico*: el cliente siempre se enruta a un punto de conexión determinado, basándose en el origen de su solicitud. Si el cliente está en Estados Unidos, siempre será dirigido al Este de EE. UU., por ejemplo. Este método de enrutamiento requiere definir las regiones geográficas que se asociarán con cada punto de conexión.

Cuando utiliza el enrutamiento geográfico, tiene un poco más de control sobre los puntos de conexión que utilizan los clientes. Puede haber razones reglamentarias que exigen que los clientes en una región determinada siempre utilicen puntos de conexión en la misma región. Los ejercicios usan puntos de conexión geográficos para mostrar un ejemplo más real, porque hay un truco para el enrutamiento geográfico: debe especificar un *perfil secundario*, no un punto de conexión directamente.

No pasará nada malo si utiliza el método de enrutamiento geográfico con puntos de conexión, pero la práctica recomendada es utilizar otro perfil de Traffic Manager para pasar el tráfico al punto de conexión final. ¿Por qué? Las regiones solo pueden asociarse con un perfil de Traffic Manager. En los capítulos anteriores sobre alta disponibilidad, siempre debió asegurarse de tener redundancia. Si asocia una región con un punto de conexión determinado y utiliza enrutamiento geográfico, no tiene ninguna opción de conmutación por error si ese punto de acceso final tiene problemas o si hace mantenimiento.

En lugar de ello, los perfiles secundarios anidados permiten establecer una prioridad que siempre dirige el tráfico a un punto de conexión en buen estado. Si el punto de conexión no está en buen estado, el tráfico se va a un punto de conexión alternativo. En la figura 11.6, se muestra el tráfico que comuta por error a una región diferente, aunque también podría crear varias instancias de aplicación web en el Oeste

de EE. UU. y utilizar un método de enrutamiento ponderado en el perfil secundario. A medida que comienza a escalar el entorno de aplicación, tómese un tiempo para pensar en la mejor manera de proporcionar alta disponibilidad a los puntos de conexión detrás de Traffic Manager. Para estos ejemplos, creará una conmutación por error entre regiones para ver claramente las diferencias de comportamiento.

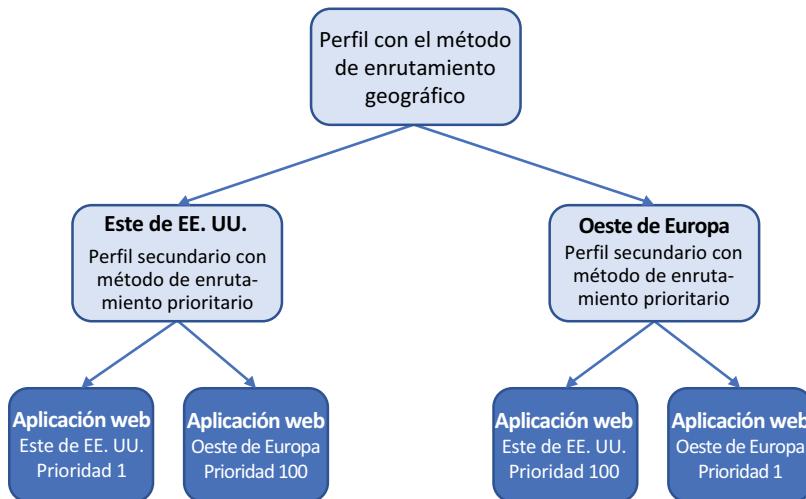


Figura 11.6 Un perfil de Traffic Manager primario con el método de enrutamiento geográfico debe utilizar perfiles secundarios que contengan varios puntos de conexión. Entonces, esos puntos de conexión secundarios pueden utilizar el enrutamiento de prioridad para dirigir el tráfico al punto de conexión preferido. Por ejemplo, el perfil secundario de Este de EE. UU. siempre y cuando el punto de conexión esté en buen estado. Si el punto de conexión no está en buen estado, el tráfico se dirige a Oeste de Europa. Sin este perfil secundario, los clientes de Este de EE. UU. no podrían conmutar por error a un punto de conexión alternativo y no podrían acceder a su aplicación web.

Pruébelo ahora

Complete los siguientes pasos para crear los perfiles de Traffic Manager para la aplicación distribuida.

El resto de los ejercicios usan Este de EE. UU. y Oeste de Europa. Si no vive en una de esas regiones, escoja una región que sea más apropiada. Solo recuerde ser coherente a lo largo de los ejercicios. El laboratorio de fin del capítulo muestra la forma en que todo esto funciona en conjunto, pero no se dirigirá correctamente a sus aplicaciones web si vive fuera de Norteamérica o Europa y no cambia las regiones como corresponde.

- 1 Abra Azure Portal y seleccione el icono Cloud Shell en la parte superior del panel.
- 2 Cree un grupo de recursos, especificando un nombre para el grupo de recursos, como `azuremolchapter11` y una ubicación, como `eastus`:


```
az group create --name azuremolchapter11 --location eastus
```
- 3 Cree el perfil de Traffic Manager primario. Utilice el método de enrutamiento geográfico y, a continuación, especifique un nombre, como `azuremol`. El parámetro del nombre DNS indica que debe ser único, así que proporcione

un nombre único. El siguiente dominio crea el nombre de host azuremol.trafficmanager.net, que se utiliza para configurar las aplicaciones web en el laboratorio de fin del capítulo:

```
az network traffic-manager profile create \
--resource-group azuremolchapter11 \
--name azuremol \
--routing-method geographic \
--unique-dns-name azuremol
```

- 4 Cree uno de los perfiles de Traffic Manager secundarios. Esta vez, utilice el método de enrutamiento prioritario y el nombre eastus, y especifique otro nombre DNS único, como azuremoleastus:

```
az network traffic-manager profile create \
--resource-group azuremolchapter11 \
--name eastus \
--routing-method priority \
--unique-dns-name azuremoleastus
```

- 5 Cree un perfil más de Traffic Manager secundario con el nombre westeurope y otro nombre DNS único, como azuremolwesterurope:

```
az network traffic-manager profile create \
--resource-group azuremolchapter11 \
--name westeurope \
--routing-method priority \
--unique-dns-name azuremolwesterurope
```

- 6 Ya ha creado una aplicación web un par de veces, así que utilicemos la CLI para crear rápidamente dos planes de App Services y luego una aplicación web en cada plan. Una de estas aplicaciones web está en Este de EE. UU. y la otra en Oeste de Europa. En el laboratorio de fin del capítulo, cargará páginas web de ejemplo a estas aplicaciones web, así que por ahora solo tiene que crear el sitio web vacío y alistarlas para usar un repositorio de Git local.

Cree la aplicación web en Este de EE. UU. de la siguiente manera:

```
az appservice plan create \
--resource-group azuremolchapter11 \
--name appserviceeastus \
--location eastus \
--sku S1
az webapp create \
--resource-group azuremolchapter11 \
--name azuremoleastus \
--plan appserviceeastus \
--deployment-local-git
```

- 7 Cree una segunda aplicación web en Oeste de Europa:

```
az appservice plan create \
--resource-group azuremolchapter11 \
--name appservicewesterurope \
--location westeurope \
--sku S1
az webapp create \
--resource-group azuremolchapter11 \
--name azuremolwesterurope \
--plan appservicewesterurope \
--deployment-local-git
```

11.3.2 Distribución global del tráfico a la instancia más cercana

Ha creado los perfiles y puntos de conexión de Traffic Manager, pero el tráfico no puede fluir. Si los clientes se redirigieran a los perfiles, no habría asociación con sus puntos de conexión. El diagrama de la figura 11.7 muestra cómo debe asociar los puntos de conexión con los perfiles.

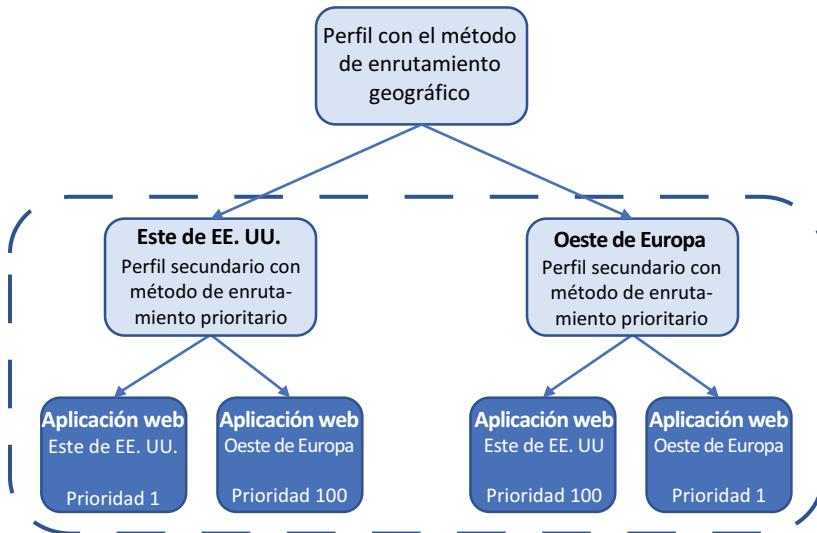


Figura 11.7 En esta sección, asociará sus puntos de conexión con los perfiles de Traffic Manager y definirá la prioridad del tráfico que se va a distribuir.

Las primeras asociaciones que realice son para sus puntos de conexión de la aplicación web. Recuerde que para una alta disponibilidad, deben estar disponibles ambas aplicaciones web para cada perfil de Traffic Manager. Utilizará un método de enrutamiento prioritario para dirigir todo el tráfico a la aplicación web principal para cada perfil. Si esa aplicación web no está disponible, el tráfico puede comutar por error al punto de conexión secundario de la aplicación web.

Cuando creó los perfiles de Traffic Manager en la sección 11.3.1, se utilizaron algunos valores predeterminados para las opciones de comprobación de estado y supervisión del punto de conexión. Exploraremos cuáles son esas opciones:

- *Período de vida (TTL) de DNS: 30 segundos.* Define el tiempo que pueden almacenarse en caché las respuestas DNS de Traffic Manager. Un TTL corto garantiza que el tráfico del cliente se enrute adecuadamente cuando se realizan actualizaciones en la configuración de Traffic Manager.
- *Protocolo de supervisión del punto de conexión: HTTP.* También puede elegir HTTPS o una comprobación TCP básica. Al igual que con los equilibradores de carga, HTTP o HTTPS garantiza que se devuelva una respuesta HTTP 200 OK de cada punto de conexión.
- *Puerto: 80.* Puerto para comprobar cada punto de conexión.

- *Ruta:* /. De forma predeterminada, comprueba la raíz del punto de conexión, aunque también puede configurar una página personalizada, como la página de comprobación de estado que utilizan los equilibradores de carga.
- *Intervalo de sondeo de puntos de conexión:* 30 segundos. La frecuencia con que se comprueba el estado del punto de conexión. El valor puede ser de 10 segundos o 30 segundos. Para realizar sondeos rápidos cada 10 segundos, hay un cargo adicional por punto de conexión.
- *Número de errores tolerables:* 3. Número de veces que un punto de conexión puede fallar una comprobación de estado antes de que el punto de conexión se marque como no disponible.
- *Número de errores tolerables:* 10 segundos. Duración antes de que un sondeo se marque como fallido y se vuelva a sondear el punto de conexión.

No es necesario cambiar ninguna de estas opciones predeterminadas. Para cargas de trabajo críticas cuando compila sus propios entornos de aplicaciones en el mundo real, se puede reducir el número de fallas a tolerar o el intervalo de sondeo. Estos cambios garantizan que cualquier problema en el estado se detecte rápidamente y el tráfico se enrute antes a otro punto de conexión.

Pruébelo ahora

Complete los siguientes pasos para asociar los puntos de conexión a los perfiles y finalizar el enrutamiento geográfico:

- 1 En Azure Portal, busque y seleccione su grupo de recursos. Para este ejercicio, seleccione el perfil de Traffic Manager que creó para Este de EE. UU.
- 2 Seleccione puntos de conexión en la barra de navegación a la izquierda del perfil y luego seleccione Agregar.
- 3 Cree un punto de conexión de Azure y escriba un nombre, como eastus.
- 4 Hay diferentes tipos de recursos de destino; desea usar App Service. Para el recurso de destino, seleccione la aplicación web en Este de EE. UU., como azuremoleastus.
- 5 Deje la prioridad en 1, acepte cualquier otro valor predeterminado que se pueda estar configurado y, a continuación, seleccione Aceptar.
- 6 Repita el proceso para agregar otro punto de conexión. Esta vez, nombre el punto de conexión westeurope, seleccione su aplicación web en Oeste de Europa como el Recurso de destino y establezca una prioridad de 100.

Ahora, su perfil de Traffic Manager enumera dos puntos de conexión: uno para la aplicación web en Este de EE. UU. y otro para la aplicación web en Oeste de Europa, como se muestra en la figura 11.8. Este enrutamiento basado en prioridades de los puntos de conexión siempre dirige el tráfico a la aplicación web en Este de EE. UU. cuando ese recurso está en buen estado. Si ese recurso no está disponible, hay redundancia para conmutar por error a la aplicación web en Oeste de Europa.

Name	Status	Monitor status	Type	Priority
eastus	Enabled	Online	Azure endpoint	1
westeurope	Enabled	Online	Azure endpoint	100

Figura 11.8 Se muestran dos puntos de conexión para el perfil de Traffic Manager. El punto de conexión para Este de EE. UU. tiene la prioridad más baja, así que siempre recibe tráfico cuando el punto de conexión está en buen estado. La redundancia se proporciona con el punto de conexión de Oeste de Europa, que solo se utiliza cuando el punto de conexión de Este de EE. UU. no está disponible.

- 7 Vuelva al grupo de recursos y seleccione el perfil de Traffic Manager para Oeste de Europa.
- 8 Elija agregar puntos de conexión.
- 9 Repita los pasos para agregar dos puntos finales y configúrelos como sigue:
 - Nombre: westeurope
Recurso de destino: aplicación web en Oeste de Europa
Prioridad: 1
 - Nombre: eastus
Recurso de destino: aplicación web en Este de EE. UU.
Prioridad: 100

Ahora, su perfil de Traffic Manager enumera dos puntos de conexión: uno para la aplicación web en Oeste de Europa y otro para la aplicación web en Este de EE. UU., como se muestra en la figura 11.9. Ha proporcionado la misma redundancia que el perfil de Traffic Manager anterior, esta vez con todo el tráfico que va a Oeste de Europa cuando está en buen estado y Este de EE. UU. si no lo está.

Name	Status	Monitor status	Type	Priority
westeurope	Enabled	Online	Azure endpoint	1
eastus	Enabled	Online	Azure endpoint	100

Figura 11.9 La misma configuración de los puntos de conexión que el perfil de Traffic Manager anterior, esta vez con la ubicación de las aplicaciones web invertidas. Estos perfiles secundarios se pueden utilizar para enrutar a los clientes a la aplicación web en Este de EE. UU. u Oeste de Europa, pero ahora tiene redundancia para conmutar por error a otro punto de conexión si el punto de conexión primario de la región no está disponible.

Solo falta una parte más para este proceso, ¡lo prometo! Recuerde, esta es una práctica recomendada para alta disponibilidad si utiliza Traffic Manager para la distribución global de aplicaciones. En el mundo real, su entorno puede no ser tan complejo. Mire el diagrama para ver los perfiles secundarios y las asociaciones con las aplicaciones web regionales que necesita crear, como se muestra en la figura 11.10.

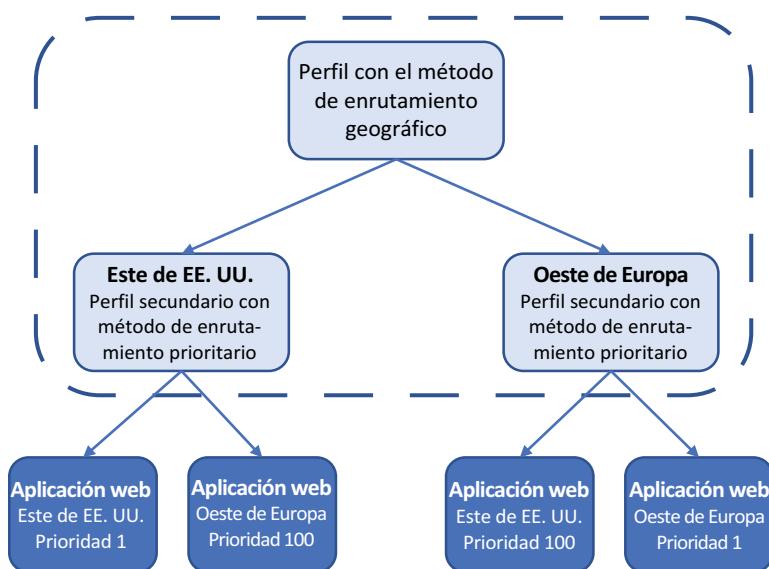


Figura 11.10 Se han creado perfiles secundarios de Traffic Manager para Este de EE. UU. y Oeste de Europa, con las prioridades y aplicaciones web regionales configuradas según sea necesario. Ahora necesita asociar los perfiles secundarios con el perfil primario.

Para dirigir el tráfico basado en la región geográfica, debe definir una región, como Norteamérica, y un perfil anidado, como eastus. Todos los clientes de la región de Norteamérica son dirigidos a este perfil secundario. Configuró las prioridades de ese perfil secundario para que la aplicación web en Este de EE. UU. siempre sirva el tráfico. Pero ha proporcionado una opción redundante para comutar por error a la aplicación web en Oeste de Europa, según sea necesario.

El inverso sucede para los clientes en Oeste de Europa. Se puede agregar otro punto de conexión para el perfil del Traffic Manager primario, esta vez con Europa como la región que se asociará con el punto de conexión y luego el perfil anidado westeurope. Todo el tráfico europeo se enruta a este perfil, y la aplicación web en Oeste de Europa siempre sirve a la aplicación web. En caso de problemas, el tráfico puede comutar por error a Este de EE. UU.

Si tiene mandatos de soberanía de directiva o de datos, tales como que el tráfico no puede comutar por error a una región diferente de esta forma, deberá ajustar la forma en que se configuran los puntos de conexión y los perfiles de Traffic Manager. Puede, por ejemplo, crear varias aplicaciones web en Oeste de Europa, como vimos en el

capítulo 9. De esta manera, tiene varias instancias de aplicación web que pueden servir a los clientes. O bien, si la aplicación se ejecuta en VM, utilice un conjunto de escalado detrás del equilibrador de carga para perfilar una redundancia similar.

Pruébelo ahora

Aquí es donde importa su propia ubicación regional. Si vive fuera de una de las agrupaciones regionales que se muestran en los perfiles de Traffic Manager, asegúrese de seleccionar su propia región o no podrá acceder a la aplicación web en el laboratorio de fin del capítulo.

Complete los siguientes pasos para asociar los perfiles secundarios con el perfil primario:

- 1 En Azure Portal, busque y seleccione su grupo de recursos.
- 2 Seleccione el perfil primario de Traffic Manager. En los ejemplos anteriores, se llamaba azuremol.
- 3 Seleccione puntos de conexión en la barra de navegación a la izquierda del perfil y luego seleccione Agregar.
- 4 Cree un punto de conexión que utilice el primer perfil secundario. Defina el tipo como un punto de conexión anidado y proporcione un nombre, como eastus. Como el recurso de destino, seleccione el perfil de Traffic Manager que creó para Este de EE. UU.
- 5 En Agrupación regional, seleccione Norteamérica/Centroamérica/Caribe en el menú desplegable y, a continuación, seleccione Aceptar.
- 6 Repita los pasos para agregar otro punto de conexión. Esta vez, nombre el punto de conexión westeurope, defina el recurso de destino en el perfil secundario de Traffic Manager para Oeste de Europa y elija Europa en el menú desplegable para agrupación regional.

Ahora, sus puntos de conexión para el perfil primario enumeran los dos perfiles secundarios, donde cada uno tiene un punto de conexión asociado con la región geográfica adecuada, como se muestra en la figura 11.11.

The screenshot shows the Azure portal interface for managing Traffic Manager profiles. The URL in the address bar is: Dashboard > Resource groups > azuremolchapter11 > azuremol - Endpoints. The main title is "azuremol - Endpoints" with a subtitle "Traffic Manager profile". On the left, there's a sidebar with links: Overview, Activity log, Access control (IAM), Tags, and Diagnose and solve problems. The main content area has a search bar "Search (Ctrl+Shift+F)" and buttons "+ Add" and "Refresh". Below is a table titled "Search endpoints" with columns: Name, Status, Monitor status, Type. The table contains two rows: "eastus" (Enabled, Online, Nested endpoint) and "westeurope" (Enabled, Online, Nested endpoint).

Figura 11.11 Perfiles secundarios anidados con regiones geográficas asociadas. Este perfil primario de Traffic Manager dirige todo el tráfico de Europa a la aplicación web en Oeste de Europa, con redundancia para usar Este de EE. UU. si hubiera algún problema. Lo contrario es verdadero para los clientes en Norteamérica/Centroamérica/Caribe.

Las aplicaciones web están actualmente configuradas para que solo acepten tráfico en su dominio predeterminado, que tiene el formato *webappname.azurewebsites.net*. Cuando Traffic Manager dirige a los clientes a las instancias de aplicaciones web, el tráfico parece provenir del dominio del perfil primario, como *azuremol.trafficmanager.net*. Las aplicaciones web no reconocen este dominio, así que la aplicación web no se cargará.

- 7 Agregue el dominio del perfil primario de Traffic Manager a las dos instancias de la aplicación web que creó en los pasos 4-6. Si fuera necesario, puede encontrar el nombre de dominio en la página de Información general del perfil primario de Traffic Manager:

```
az webapp config hostname add \
--resource-group azuremolchapter11 \
--webapp-name azuremoleastus \
--hostname azuremol.trafficmanager.net
az webapp config hostname add \
--resource-group azuremolchapter11 \
--webapp-name azuremolwesteurope \
--hostname azuremol.trafficmanager.net
```

Ahora, cuando abra la dirección de su perfil principal de Traffic Manager en un navegador web, como por ejemplo <https://azuremol.trafficmanager.net>, no se puede saber a qué punto de conexión accede, ya que ambas aplicaciones web ejecutan la página web predeterminada. En el laboratorio de fin del capítulo, cargará una página web básica a cada aplicación web para diferenciarlas.

Detengámonos para examinar lo que ha creado con estos ejercicios. Es importante, porque ahora los clientes pueden usar todas las funcionalidades de alta disponibilidad y redundancia de capítulos anteriores, con enrutamiento de tráfico automático que los dirige a la instancia más cercana de su aplicación web. En este capítulo, ha creado lo siguiente:

- Una aplicación web en Este de EE. UU. y otra en Oeste de Europa.
- Perfiles de Traffic Manager que utilizan enrutamiento geográfico para dirigir a todos los clientes de Norteamérica y Centroamérica a la aplicación web de Este de EE. UU. y a todos los clientes en Europa a la aplicación web de Oeste de Europa.
- Directivas secundarias de Traffic Manager con enrutamiento prioritario para usar mediante conmutación por error la región alternativa si la aplicación web principal de la región no está disponible.

En términos de alta disponibilidad:

- Si combina esta configuración con aplicaciones web que escalan automáticamente, tiene un montón de redundancia en este momento.
- Si combina estas aplicaciones web con Cosmos DB, ahora toda su aplicación escala de manera automática y está distribuida globalmente, los clientes siempre acceden a los recursos cercanos a ellos para una latencia más baja en los tiempos de respuesta y un mejor rendimiento.
- Incluso si se enreda con las VM, puede utilizar conjuntos de escalado con equilibradores de carga para proporcionar el mismo entorno altamente disponible y distribuido de manera global.

Y sí, podría reemplazar Traffic Manager con Front Door si necesita utilizar funciones avanzadas de administración de tráfico en el nivel de aplicación.

Sé que los últimos capítulos contienen un montón de cosas nuevas y cada capítulo le ha tomado prácticamente todo su descanso para almorzar a diario. Pero mire hasta dónde llegó la semana pasada. Ahora puede crear una aplicación web con VM IaaS o aplicaciones web PaaS, hacerlas altamente disponibles con carga equilibrada, y dejarlas escalar de manera automática (figura 11.12). Puede utilizar una base de datos de Cosmos DB de back-end distribuida globalmente para sus necesidades de bases de datos y puede enrutar de manera automática a los clientes a la instancia regional más cercana de su aplicación, todo con DNS alojado en Azure.

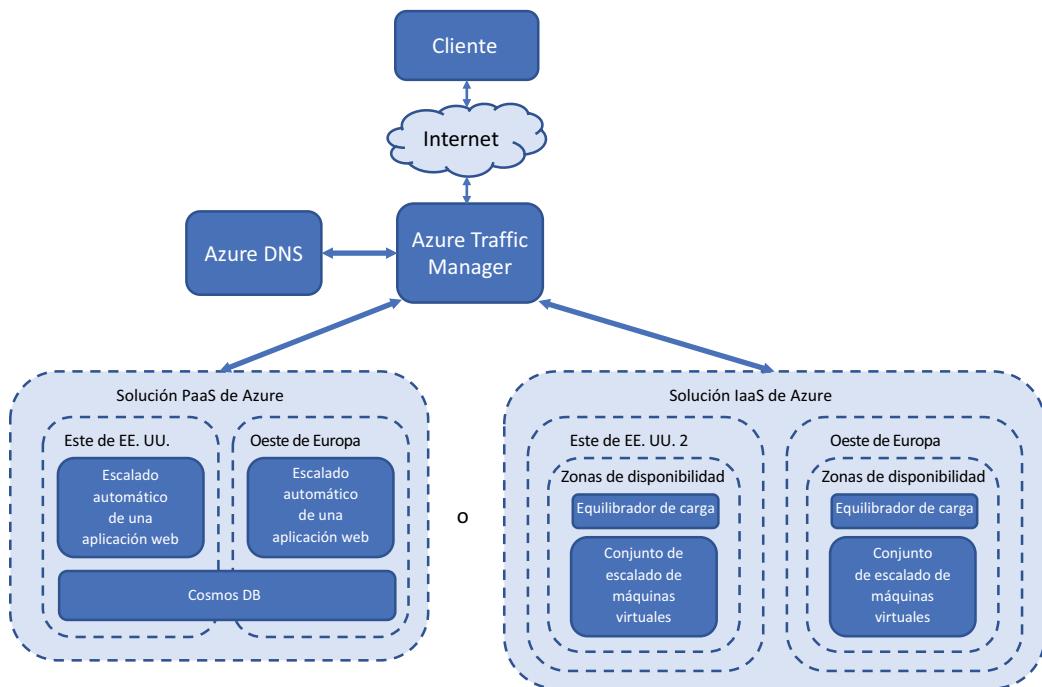


Figura 11.12 Despues de los últimos capítulos, entenderá cómo crear aplicaciones IaaS o PaaS altamente disponibles en Azure. Las soluciones IaaS pueden utilizar zonas de disponibilidad, equilibradores de carga y conjuntos de escalado. Las soluciones PaaS pueden utilizar aplicaciones web de escalado automático y Cosmos DB. Traffic Manager y Azure DNS pueden enrutar automáticamente a los clientes a la instancia de aplicación más adecuada, basándose en su ubicación geográfica.

En el laboratorio de fin del capítulo cargará un par de sitios web básicos a sus aplicaciones web, solo para demostrar que Traffic Manager funciona y que el punto de conexión adecuado sirve su tráfico. Si tiene tiempo, puede completar el ejercicio; de lo contrario, relájese. ¡No se lo diremos a su jefe!

Tenemos un capítulo más en esta segunda sección del libro, que trata sobre cómo asegurarse de que sus aplicaciones permanezcan en buen estado: cómo supervisar y solucionar los problemas de sus aplicaciones e infraestructura.

11.4 Laboratorio: Implementación de aplicaciones web para ver Traffic Manager en acción

Este ha sido otro capítulo donde hemos pasado mucho contenido, así que este ejercicio debiera seguir compilando musculatura mental de sus habilidades de Azure con aplicaciones web. En el repositorio de ejemplos de GitHub de Azure hay dos páginas web básicas para la aplicación de pizzería en línea. El título de cada página web muestra la ubicación de la aplicación web. Cargue estas páginas web en la instancia pertinente de la aplicación web para ver los flujos de Traffic Manager en práctica:

- 1 Si fuera necesario, clone el repositorio de ejemplos de GitHub en su Cloud Shell de la siguiente manera:

```
git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
```

- 2 Comience con la página web eastus y repita los pasos siguientes en el directorio westeurope:

```
cd ~/azure-mol-samples-2nd-ed/11/eastus
```

- 3 Inicialice el repositorio Git y agregue la página web básica:

```
git init && git add . && git commit -m "Pizza"
```

- 4 En Azure Portal, la URL de Git Clone aparece en la ventana de Información general de su aplicación web. Copie esta URL y, a continuación, defínala como destino para el sitio HTML de ejemplo en Cloud Shell con el siguiente comando:

```
git remote add eastus <your-git-clone-url>
```

- 5 Inserte el sitio HTML de ejemplo en su aplicación web:

```
git push eastus master
```

- 6 Repita estos pasos para el directorio azure-mol-samples-2nd-ed/11/westeurope.

- 7 Cuando haya terminado, abra el navegador web con el nombre de dominio del perfil de Traffic Manager, como <https://azuremol.trafficmanager.net>, para ver el flujo de tráfico.

Monitoreo y solución de problemas



En los capítulos anteriores, aprendió a hacer que sus aplicaciones estuvieran altamente disponibles y a enrutar clientes de todo el mundo a instancias distribuidas globalmente de su aplicación. Un objetivo era minimizar la cantidad de interacción con la infraestructura de su aplicación y permitir que la plataforma Azure administrara automáticamente el estado y el rendimiento. A veces, todavía necesita subirse las mangas y revisar los diagnósticos o métricas de rendimiento. En este capítulo, aprenderá a revisar los diagnósticos de arranque para una VM, supervisar las métricas de rendimiento y solucionar problemas de conectividad con Network Watcher.

12.1 Diagnóstico de arranque de VM

Con Web Apps, usted implementa el código y deja que la plataforma Azure se encargue del resto. En el capítulo 3, analizamos los fundamentos de cómo solucionar y diagnosticar el problema de las implementaciones de aplicaciones web. Aprendió a ver los eventos de aplicación en tiempo real para supervisar el rendimiento. Cuando trabaja con VM en la nube, suele ser difícil solucionar un problema cuando no se puede ver físicamente la pantalla del equipo de la manera en que ve los diagnósticos de aplicaciones web.

Uno de los problemas más comunes con las VM es la falta de conectividad. Si no puede aplicar SSH o RDP a una VM, ¿cómo puede solucionar el problema? Una de las primeras cosas que debe comprobar es si la VM se está ejecutando correctamente. Para ayudarlo a hacer esto, Azure ofrece diagnósticos de arranque de VM que incluye registros de arranque y una captura de pantalla de la consola.

Acceso interactivo de la consola de arranque

Para situaciones específicas de solución de problemas, también puede acceder a una consola serie para VM en vivo en Azure. Esta consola serie permite los arranques de sesión interactivos y la solución de problemas de arranque. Puede volver a configurar su VM para corregir situaciones de arranque fallidos o configuraciones erróneas de servicios y aplicaciones que impiden que su VM se inicie correctamente.

Este capítulo no abarca situaciones específicas para el uso de la consola serie, pero es un gran recurso que le permite prácticamente sentarse frente a la pantalla de una VM mientras se inicia. También necesita habilitar los diagnósticos de arranque, así que estos ejercicios son un prerequisitos para la consola serie.

Pruébelo ahora

Complete los siguientes pasos para crear una VM y habilitar el diagnóstico de arranque:

- 1 En Azure Portal, seleccione Crear un recurso en la esquina superior izquierda.
- 2 Busque y seleccione una imagen de máquina virtual de Windows Server 2019 Datacenter.
- 3 Cree un grupo de recursos, como azuremolchapter12 y, luego, seleccione la región de Azure correspondiente más cercana a usted.
- 4 Seleccione un tamaño de VM, como DS1_v2.
- 5 Escriba un nombre de usuario para la VM, como azuremol, y una contraseña. La contraseña debe tener un mínimo de 12 caracteres y contener 3 de los siguientes: un carácter en minúscula, un carácter en mayúscula, un número y un carácter especial.
- 6 Acepte cualquier opción de redundancia o reglas de puerto de entrada.
- 7 Acepte los valores predeterminados para los discos y las redes; no hay nada que necesite cambiar. Esa configuración tiene que ser familiar para usted por ahora.

Una sección que se puede haber saltado anteriormente es la sección Administración. Como se muestra en la figura 12.1, la opción de diagnóstico de arranque está activada de forma predeterminada y se crea una cuenta de almacenamiento.

- 8 Por ahora, deje desactivada la opción de diagnóstico del sistema operativo invitado.
- 9 Revise la configuración de la máquina virtual y seleccione Crear.

La VM demora unos minutos en crearse y configurarse, así que continuemos explorando los diagnósticos de arranque.

Si no tiene habilitado el diagnóstico de arranque pero se produce un problema, es probable que no pueda iniciar la VM para habilitar correctamente los diagnósticos. Es como el caso del huevo y la gallina, ¿verdad? Como resultado, los diagnósticos de arranque se habilitan automáticamente para las VM creadas en Azure Portal. Para Azure PowerShell, la CLI de Azure y los SDK específicos de cada lenguaje,

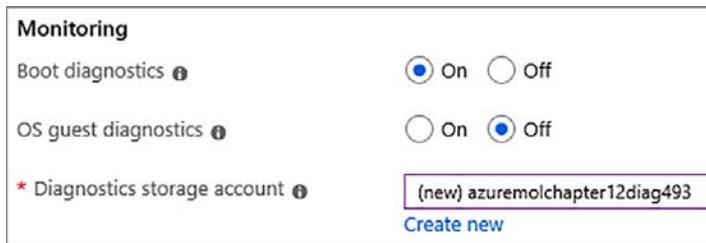


Figura 12.1 De forma predeterminada, los diagnósticos de arranque se habilitan cuando se crea una VM en Azure Portal. Se crea una cuenta de almacenamiento; los diagnósticos de arranque se almacenan en esta cuenta. En un ejercicio posterior, revisará y habilitará los diagnósticos del sistema operativo invitado, así que no los habilite en este momento. Para producción, le recomendamos habilitar tanto los diagnósticos de arranque como los diagnósticos de SO invitado para cada VM que cree.

es necesario habilitar el diagnóstico de arranque. Le recomiendo encarecidamente que habilite los diagnósticos de arranque en las VM al crearlos. Acostúmbrese a utilizar las plantillas de Azure Resource Manager (capítulo 6) o sus propios scripts de CLI de Azure o PowerShell que permiten el diagnóstico de arranque durante la implementación.

Es necesario crear una cuenta de almacenamiento para los registros de arranque y las capturas de pantalla de la consola, pero el costo de almacenar estos datos es probablemente inferior a USD 0,01 al mes, a menos que tenga una máquina virtual muy ocupada que genere muchos datos. La primera vez que se produzca un problema con la VM y necesite acceder a los diagnósticos de arranque, ese centavo al mes valdrá la pena. Esta cuenta de almacenamiento también se puede utilizar para contener métricas y registros adicionales de rendimiento en el nivel de la VM, que examinaremos en la sección 12.2. De nuevo, los costos de almacenamiento tienen que ser mínimos. Incluso a medida que crezca su entorno de VM, vale la pena el pequeño costo adicional para poder solucionar rápidamente un problema cuando las cosas salen mal.

Pruébelo ahora

Para ver el diagnóstico de arranque para su VM, complete los siguientes pasos:

- 1 En Azure Portal, seleccione Máquinas virtuales en el menú de la izquierda.
- 2 Elija la VM que creó en el ejercicio anterior.
- 3 En la sección Soporte técnico y solución de problemas del menú de la VM, seleccione Diagnóstico de arranque. Aparecerá el diagnóstico de arranque y el estado de la VM, como se muestra en la figura 12.2. El informe de estado indica cualquier problema de arranque con la VM y le permite diagnosticar la causa raíz del problema.

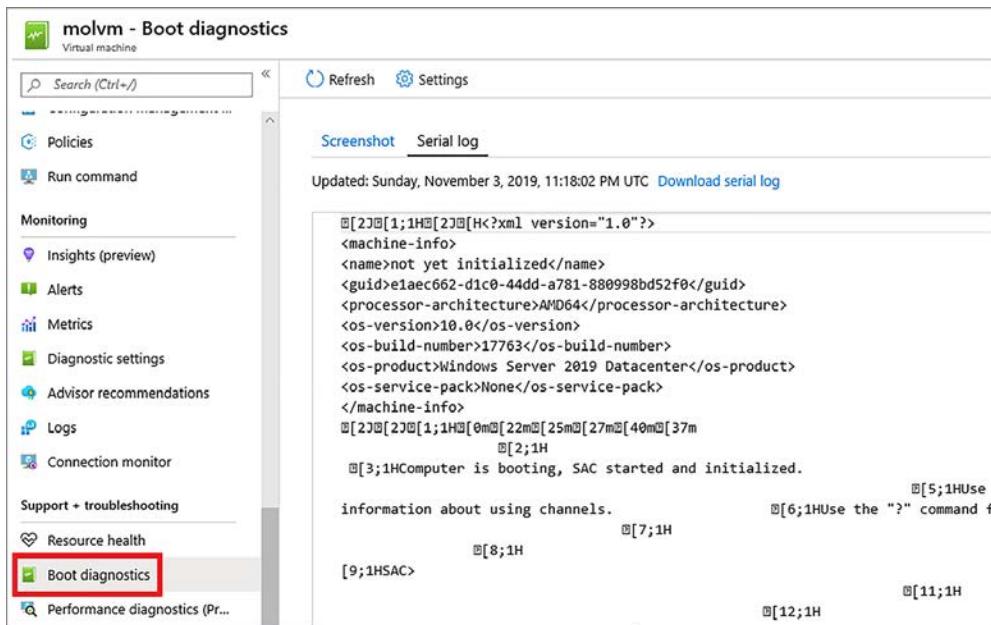


Figura 12.2 Diagnóstico de arranque de un informe de VM sobre el estado de mantenimiento y de arranque. Si se muestran errores, debería ser capaz de solucionar los problemas y diagnosticar la causa raíz. También puede descargar los registros del portal para analizarlos en el equipo local.

12.2 Métricas y alertas de rendimiento

Uno de los primeros pasos para solucionar un problema es la revisión del rendimiento. ¿Cuánta memoria hay disponible, cuánta CPU se consume y cuánta actividad de disco hay?

A medida que compile y pruebe sus aplicaciones en Azure, le recomendamos registrar las líneas base de rendimiento en varios puntos. Estas líneas base le dan una idea de cómo debe funcionar su aplicación bajo diferentes cantidades de carga. ¿Por qué es tan importante? En tres meses, ¿cómo se puede determinar si hay problemas de rendimiento sin algunos datos para compararlos con el rendimiento actual?

Cuando aprendió a escalar automáticamente las aplicaciones en el capítulo 9, utilizó métricas básicas de rendimiento, como el uso de la CPU, para decirle a la plataforma Azure cuándo aumentar o disminuir el número de instancias de la aplicación. Estas métricas básicas solo le dan una pequeña idea de cómo funciona la VM. Para obtener métricas más detalladas, tiene que observar el rendimiento de la máquina virtual y, para ello, debe instalar la extensión de diagnósticos de Azure.

12.2.1 Visualización de métricas de rendimiento con la extensión de diagnóstico de VM

Para agregar funcionalidad a las VM, Azure tiene docenas de extensiones que puede instalar sin problemas. Estas extensiones instalan un pequeño agente o tiempo de ejecución de las aplicaciones en la VM que suele reportar la información a la plataforma Azure o a soluciones de terceros. Las extensiones de VM pueden configurar e instalar componentes automáticamente o ejecutar scripts en las VM.

La extensión de diagnóstico de VM es una común que se utiliza para transmitir métricas de rendimiento desde el interior de la VM a una cuenta de almacenamiento. Estas métricas de rendimiento pueden ser analizadas en Azure Portal, o descargadas y usadas en una solución de supervisión existente. Puede utilizar la extensión de diagnóstico para obtener una comprensión más profunda del rendimiento de la CPU y el consumo de memoria dentro de la VM, que normalmente puede proporcionar un panorama más detallado y preciso que el host.

Automatización y extensiones de VM

En el capítulo 18, hablamos de Azure Automation, que le permite realizar tareas de forma automatizada y programada en la VM. Una característica poderosa de Azure Automation es actuar como un servidor de extracción de Desired State Configuration (DSC) de PowerShell. PowerShell DSC define un estado determinado de cómo debe configurarse un sistema, qué paquetes deben instalarse, archivos y permisos, etc. Usted crea definiciones para la configuración deseada y se aplican a las VM o servidores físicos. A continuación, puede informar y aplicar el cumplimiento de esas directivas. La extensión de Azure PowerShell DSC se utiliza para aplicar configuraciones DSC, como por ejemplo, desde un servidor de extracción de Azure Automation.

Otras extensiones que pueden aplicar configuraciones y ejecutar scripts en VM incluyen Azure Custom Script Extension. Con la extensión de la secuencia de comandos personalizada, puede definir un conjunto simple de comandos o dirigir a uno o más scripts externos, como los alojados en Azure Storage o GitHub. Estos scripts pueden ejecutar tareas complejas de instalación y configuración, y garantizar que todas las VM implementadas estén configuradas de manera uniforme.

La extensión de Azure PowerShell DSC y la extensión de la secuencia de comandos personalizada se utilizan comúnmente con conjuntos de escalado de máquinas virtuales. Se aplica una de estas extensiones al conjunto de escalado y, a continuación, cuando se crean instancias de VM dentro del conjunto de escalado, se configuran automáticamente para ejecutar la aplicación. El objetivo de estas extensiones es minimizar la configuración manual requerida de las VM, que es un proceso propenso a errores y que requiere interacción humana.

Otras formas de automatizar las configuraciones de VM incluyen Puppet y Chef, ambos con extensiones de la VM Azure disponibles. Si ya tiene una herramienta de administración de configuración en uso, consulte con su proveedor el enfoque compatible para utilizar en Azure. Es muy probable que haya una extensión de VM disponible para hacer su vida más fácil.

Pruébelo ahora

Complete los siguientes pasos para habilitar la extensión de diagnóstico de VM:

- 1 En Azure Portal, seleccione Máquinas virtuales en el menú de la izquierda.
- 2 Elija la VM que creó en el ejercicio anterior.
- 3 En la sección Monitor del menú de VM, seleccione Configuración de diagnóstico.
- 4 Seleccione el botón para activar la supervisión en el nivel de invitado.

La supervisión de nivel de invitado tarda un par de minutos en activarse. Esto es lo que hace Azure:

- Instala la extensión de diagnóstico de VM
- Configura la extensión para transmitir métricas de nivel de invitado para las siguientes áreas:
 - Disco lógico
 - Memoria
 - Interfaz de red
 - Procesos
 - Procesador
 - Sistema
- Habilita la aplicación, la seguridad y los registros del sistema para que se transmitan a Azure Storage

Cuando instale la extensión de diagnóstico, puede limitar los datos que se recopilan seleccionando solo ciertos contadores de rendimiento a reportar. Por ejemplo, es posible que desee recopilar solo el uso de memoria o habilitar la recopilación de métricas de Microsoft SQL Server. De forma predeterminada, las métricas se recopilan cada 60 segundos. Puede ajustar esta velocidad de muestreo como desee para sus aplicaciones e infraestructura.

La extensión de diagnóstico de VM también puede transmitir archivos de registro de su máquina virtual, lo que le permite centralizar los registros de la aplicación, la seguridad y el sistema para el análisis o las alertas, como se muestra en la figura 12.3. De forma predeterminada, se registran los sistemas y los registros de aplicaciones que generan alertas críticas, errores o advertencias, junto con eventos de seguridad para las auditorías con errores. Puede cambiar los niveles de registro para registrar, así como también habilitar la recopilación de registros de IIS, los registros de aplicaciones y los eventos de Seguimiento de eventos para Windows (ETW). Como parte de la planificación e implementación de aplicaciones, determine qué registros desea recopilar.

Aquí no hay nada único de las VM Windows. Puede utilizar la extensión de diagnóstico en las VM Linux de la misma manera, a fin de obtener métricas de rendimiento y transmitir varios registros.

Por lo general, si su VM encuentra un problema, la única manera de analizar lo que sucedió es revisando los *volcados de memoria*. Los canales de soporte suelen solicitar estos volcados si desea llegar a la causa raíz de un problema. Al igual que con los diagnósticos de arranque, no hay manera de habilitar retroactivamente los volcados de memoria para ver por qué algo falló, así que determine si necesita supervisar ciertos procesos y sea proactivo con la configuración de los volcados de memoria. Por ejemplo, puede supervisar el proceso de IIS y registrar un volcado de memoria en Azure Storage si el proceso falla.

Aquí hay un par de otras áreas que puede configurar para las métricas de invitado:

- *Los receptores* le permiten configurar la extensión de diagnóstico de VM para enviar ciertos eventos a Azure Application Insights. Con Application Insights, puede obtener visibilidad directa sobre cómo funciona el código.
- *El agente* le permite especificar una cuota de almacenamiento para todas sus métricas (El valor predeterminado es 5 GB). También puede habilitar la recolección de registros para el propio agente o desinstalar el agente.

The screenshot shows the 'Logs' tab in the Azure portal's Metrics and Alerts section. It includes configuration for event logs, IIS logs, and failed request logs. For event logs, there are three main sections: Application, Security, and System. Each section has checkboxes for Critical, Error, Warning, Information, and Verbose levels. For IIS logs and failed request logs, there are fields for 'Storage container name'.

Figura 12.3 Puede configurar eventos y niveles de registro para varios componentes dentro de la VM. Esta característica permite centralizar los registros de VM para análisis y generación de alertas. Podrá revisar y recibir notificaciones cuando surjan problemas en sus VM de Azure, sin necesidad de instalar sistemas de supervisión complejos, que suelen ser costosos.

Pruébelo ahora

Complete los siguientes pasos para ver las métricas en el nivel de invitado:

- 1 En Azure Portal, seleccione Máquinas virtuales en el menú de la izquierda.
- 2 Elija la VM que creó en el ejercicio anterior.
- 3 En la sección Monitor del menú de VM, seleccione Métricas.

Ahora hay muchas más métricas disponibles, en comparación con las métricas básicas basadas en el host del capítulo 9. Explore algunas de las métricas del host y del invitado de la máquina virtual disponibles, y piense en algunas aplicaciones para las que podría querer monitorear métricas específicas.

12.2.2 Creación de alertas para condiciones de rendimiento

Con su VM configurada para exponer métricas de rendimiento en el nivel de invitado, ¿cómo sabe cuándo hay un problema? No quiere sentarse y mirar los gráficos de rendimiento en tiempo real y esperar hasta que ocurra un problema. Pero si quiere hacerlo, bienvenido. Pero hay una manera mucho mejor: las alertas de métricas.

Las alertas de métricas le permiten seleccionar un recurso, una métrica y un umbral y, a continuación, definir a quién y cómo desea notificar cuando se cumple ese umbral. Las alertas no funcionan exclusivamente en las VM. Puede definir alertas en direcciones IP públicas que detectan paquetes de denegación de servicio distribuido (DDoS) entrantes, por ejemplo, y advertirle cuando se cumple un determinado umbral que pueda constituir un ataque.

Cuando se generan alertas, puede elegir enviar una notificación por correo electrónico a dueños, colaboradores y lectores. Estos usuarios y direcciones de correo electrónico se obtienen con base en las directivas RBAC aplicadas. En organizaciones más grandes, las alertas podrían enviar notificaciones por correo electrónico a un grupo grande de personas, ¡así que úselas con cuidado! Otra opción es especificar direcciones de correo electrónico, como las de los dueños de aplicaciones o ingenieros de infraestructura específicos, o una lista de distribución o un grupo dirigido a las partes directamente involucradas.

Existe un par de otras opciones útiles para las acciones que se deben tomar cuando se activa una alerta:

- *Ejecutar un runbook.* En el capítulo 18, examinaremos Azure Automation. El servicio de automatización le permite crear y utilizar runbooks que ejecutan scripts. Estos scripts pueden realizar una acción correctiva básica en la VM, por ejemplo, para reiniciar un proceso o reiniciar la VM. También podrían ejecutar los cmdlets de Azure PowerShell para habilitar las funciones de Azure Network Watcher, como los paquetes de captura, que exploraremos en el resto de este capítulo.
- *Ejecutar una aplicación lógica.* Azure Logic Apps le permite generar flujos de trabajo que ejecuten código sin servidor. Puede escribir información en un sistema de tickets de soporte o iniciar una llamada telefónica automatizada a un ingeniero de turno. En el capítulo 21, exploraremos el maravilloso mundo de la informática sin servidor con Azure Logic Apps y Azure Functions.

En el laboratorio de fin del capítulo, configurará algunas alertas para su VM. Sin embargo, Azure puede hacer más que ayudar a solucionar problemas y supervisar sus VM. Analicemos otra causa común cuando hay problemas: la red.

12.3 **Azure Network Watcher**

Las métricas de rendimiento y los diagnósticos de arranque de VM son excelentes maneras de supervisar sus aplicaciones de IaaS Azure. Los registros de aplicaciones de Web Apps y App Insights reconocen el rendimiento de sus aplicaciones PaaS. El tráfico de red suele ser menos glamoroso, pero es más probable que sea la causa de sus problemas (o de sus clientes) de conectividad de las aplicaciones.

En el capítulo 5, bromeamos con que el equipo de red siempre es el culpable de los problemas que el equipo de operaciones no puede explicar. Aquí es donde podemos tratar de volver a hacernos amigos o al menos obtener alguna prueba sólida de que la culpable es la red. Azure Network Watcher es una de esas funciones que ayuda a unir equipos en un amistoso abrazo grupal. Con Network Watcher, puede supervisar y solucionar problemas usando características tales como:

- Captura de paquetes de red
- Validación del flujo de IP para NSG
- Generación de topología de red

Lo genial acerca de estas funciones es que ponen diferentes equipos en el asiento del conductor para solucionar problemas. Si crea algunas VM y luego no puede conectarse a ellas, puede comprobar que haya conectividad de red. Para los desarrolladores, si la

aplicación no puede conectarse a un nivel de la base de datos back-end, puede examinar las reglas de NSG para ver si hay algún problema. Y los ingenieros de red pueden capturar paquetes para examinar la transmisión de comunicación completa entre los hosts para un análisis más profundo.

Solución de problemas de red adicionales

Network Watcher trabaja en conjunto con los registros de diagnóstico y las métricas analizadas anteriormente. Los recursos de red como los equilibradores de carga y los Application Gateways también pueden generar registros de diagnóstico. Estos registros funcionan de la misma manera que los registros de aplicaciones y sistemas desde una VM o una aplicación web. Los registros se cotejan en Azure Portal para que usted determine si hay errores en la configuración o en las comunicaciones entre hosts y aplicaciones.

DNS y Traffic Manager también tienen un área de Solución de problemas en Azure Portal. El portal lo guía a través de algunos errores comunes que puede encontrar, ofrece consejos de configuración y proporciona vínculos a documentación adicional. Si todo lo demás falla, puede abrir una solicitud de asistencia para el Soporte técnico de Azure.

Aunque a menudo puede ser más fácil compilar implementaciones de aplicaciones de gran tamaño con las plantillas de Azure Resource Manager o con los scripts de CLI de Azure o PowerShell, Azure Portal cuenta con muchas herramientas y funciones excelentes que puede usar cuando está teniendo problemas. Especialmente con las configuraciones de red complicadas y las directivas de seguridad, unos pocos segundos de su tiempo para revisar los resultados de las herramientas de Network Watcher pueden ayudar a identificar un problema y resolverlo rápidamente. Todas estas herramientas ayudan a mejorar el estado general y la experiencia de las aplicaciones para sus clientes.

¿Cuáles son algunas situaciones en las que es posible utilizar Network Watcher y la función de solución de problemas que ofrece? Echemos un vistazo a algunos problemas comunes y veamos cómo Network Watcher podría ayudar.

12.3.1 Verificación de flujos de IP

Este es un problema común: los clientes no pueden conectarse a su aplicación. La aplicación funciona bien cuando se conecta desde la oficina, pero los clientes no pueden acceder a la aplicación a través de Internet público. ¿Por qué?

VPNs y ExpressRoute

Azure Virtual Private Networks (VPN) proporciona comunicaciones seguras entre las oficinas locales y los centros de datos de Azure. Azure ExpressRoute proporciona conexiones privadas de alta velocidad y exclusiva desde oficinas locales hasta los centros de datos Azure y suele utilizarse en grandes organizaciones.

Ambaras conexiones son un poco más complicadas de configurar que lo que podemos cubrir en un solo descanso de almuerzo y, por lo general, se configuran una sola vez. El equipo de red suele ser el responsable de configurarlas, y es posible que usted ni siquiera sepa que accede a Azure a través de una conexión privada.

Todas las pruebas de su aplicación funcionan bien. Puede acceder a la aplicación a través de un navegador web, realizar pedidos y recibir notificaciones por correo electrónico. Pero cuando sus clientes intentan hacer un pedido, la aplicación no se carga.

¿Cómo puede ayudar Network Watcher? Verificando los flujos IP. Network Watcher simula el flujo de tráfico a su destino e informa si el tráfico puede llegar a su VM.

Pruébelo ahora

Complete los siguientes pasos para habilitar Network Watcher y verificar flujos IP:

- 1 En Azure Portal, elija Todos los recursos en la parte superior del menú de navegación a la izquierda.
- 2 Filtre y seleccione Network Watcher en la lista de servicios disponibles. Habilite Network Watcher en las regiones que desea supervisar. Cuando habilita Network Watcher en una región, Azure utiliza controles de acceso basados en roles para los diversos recursos y tráfico de red.
- 3 Amplíe la lista de regiones para su cuenta. Es posible que algunas regiones ya estén habilitadas. Si la región en la que se implementó su máquina virtual no está habilitada, seleccione la región y, a continuación, habilite Network Watcher.
- 4 Cuando Network Watcher esté activado en una región (tarda uno o dos minutos), seleccione Comprobación del flujo de IP en Herramientas de diagnóstico de red en el lado izquierdo de la ventana de Network Watcher.
- 5 Seleccione el grupo de recursos, como azuremolchapter12 y la VM, como molvm. De forma predeterminada, el protocolo se establece en TCP y la dirección es entrante. También se rellena la dirección IP local de la NIC virtual.
- 6 Para Puerto local, ingrese puerto 80. Si aceptó los valores predeterminados al crear la VM en el ejercicio anterior, no abrió el puerto 80, así que esta es una buena prueba de lo que sucede cuando se deniega el tráfico.
- 7 En Dirección IP remota, ingrese 8.8.8.8. Esta dirección puede parecer familiar: es un servidor DNS abierto proporcionado por Google. No está haciendo nada con este servidor; solo tiene que dar a Network Watcher una dirección IP externa para simular el flujo de tráfico. También puede ir a <https://whatsmyip.com> e ingresar su dirección IP pública real.
- 8 Establezca el puerto remoto en el puerto 80 y, a continuación, seleccione Comprobar.

El resultado de la comprobación del flujo de IP debe ser Acceso denegado. Prácticamente, Network Watcher le dice qué regla provocó que el flujo de tráfico fallara: la regla DenyAllInBound. Como ya lo sabe, hay una regla de seguridad de red que bloquea el tráfico, pero ¿dónde se aplica esta regla? ¿En la subred, la NIC virtual o el grupo de seguridad de aplicaciones? Otra función de Network Watcher puede decírselo.

12.3.2 Visualización de reglas efectivas de NSG

Las reglas NSG se pueden aplicar a una única NIC virtual, en el nivel de subred o a un grupo de VM en un grupo de seguridad de aplicaciones. Las reglas se combinan, lo que permite especificar un conjunto de reglas comunes en toda una subred y luego obtener

más detalles para los grupos de seguridad de aplicaciones (como "Permitir el puerto TCP 80 en todos los servidores web") o de una VM individual.

A continuación se muestran algunos ejemplos comunes de cómo se pueden aplicar las reglas de NSG:

- *Nivel de subred*: permite que el puerto TCP 5986 administre de forma remota y segura la subred de administración 10.1.10.20/24.
- *Nivel de grupo de seguridad de aplicaciones*: permite el puerto TCP 80 para tráfico HTTP a aplicaciones web y aplica el grupo de seguridad de aplicaciones a todas las VM de aplicaciones web.
- *Nivel de NIC virtual*: permite el puerto TCP 3389 el acceso a escritorio remoto desde la subred de administración 10.1.10.20/24.

Estas reglas son básicas y permiten explícitamente cierto tráfico. Si no hay reglas *allow* que permitan coincidir con un paquete de red, se aplican las reglas *DenyAll* predeterminadas para detener el tráfico.

Durante la prueba de la aplicación analizada en el ejemplo, es posible que haya configurado esa regla HTTP solo para permitir el tráfico de una de las subredes locales. Ahora, los clientes no pueden conectarse a través de Internet público.

Pruébelo ahora

Complete los siguientes pasos para determinar dónde se aplica una regla de NSG:

- 1 En Network Watcher, seleccione Reglas de seguridad eficaces a la izquierda.
- 2 Seleccione el grupo de recursos, como azuremolchapter12 y su VM, como molvm. Las reglas efectivas tardan unos segundos en aparecer, como se muestra en la figura 12.4.

NAME	PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL	ACCESS
AllowVnetInBound	65000	Virtual network (2 prefixes)	0-65535	Virtual network (2 prefixes)	0-65535	All	Allow
AllowAzureLoadBalancer...	65001	Azure load balancer (1 prefix)	0-65535	0.0.0.0/0,0.0.0.0/0	0-65535	All	Allow
DenyAllInBound	65500	0.0.0.0/0,0.0.0.0/0	0-65535	0.0.0.0/0,0.0.0.0/0	0-65535	All	Deny
NAME	PRIORITY	SOURCE	SOURCE PORTS	DESTINATION	DESTINATION PORTS	PROTOCOL	ACCESS
AllowVnetOutBound	65000	Virtual network (2 prefixes)	0-65535	Virtual network (2 prefixes)	0-65535	All	Allow
AllowInternetOutBound	65001	0.0.0.0/0,0.0.0.0/0	0-65535	Internet (216 prefixes)	0-65535	All	Allow
DenyAllOutBound	65500	0.0.0.0/0,0.0.0.0/0	0-65535	0.0.0.0/0,0.0.0.0/0	0-65535	All	Deny

Figura 12.4 Cuando selecciona una VM, Network Watcher examina cómo se aplican todas las reglas de NSG y el orden de precedencia, y muestra qué reglas efectivas se aplican actualmente. Luego, puede examinar en profundidad rápidamente la subred, la NIC virtual y las reglas predeterminadas para buscar y editar dónde se aplica una regla determinada.

Las reglas predeterminadas de la VM creada anteriormente no son interesantes, pero puede desplazarse a través de la subred, la interfaz de red y las reglas predeterminadas para tener una idea de cómo se combinan las reglas efectivas y cómo puede identificar dónde se aplican las reglas si necesita realizar cambios.

12.3.3 Captura de paquetes de red

Supongamos que ha actualizado las reglas de seguridad de red para permitir el acceso a su aplicación para clientes de Internet público, pero un cliente informa que está experimentando un comportamiento extraño. A veces, la aplicación web no se carga o muestra imágenes incompletas. Pareciera que se agota el tiempo de espera de la conexión.

Los problemas intermitentes son a menudo los más difíciles de solucionar, especialmente si tiene acceso limitado o nulo al equipo que sufre el problema. Un enfoque de solución de problemas común es capturar los paquetes de red y revisarlos para detectar signos de cualquier problema, como errores de transmisión de red, paquetes malformados o problemas de protocolo y comunicación.

Con las capturas de paquetes de red, obtiene la transmisión de datos sin procesar entre dos o más hosts. Hay un arte para analizar las capturas de red y no es para los débiles de corazón. Herramientas especiales de terceros como Wireshark de Riverbed, Fiddler de Telerik y Message Analyzer de Microsoft proporcionan una forma gráfica para que vea

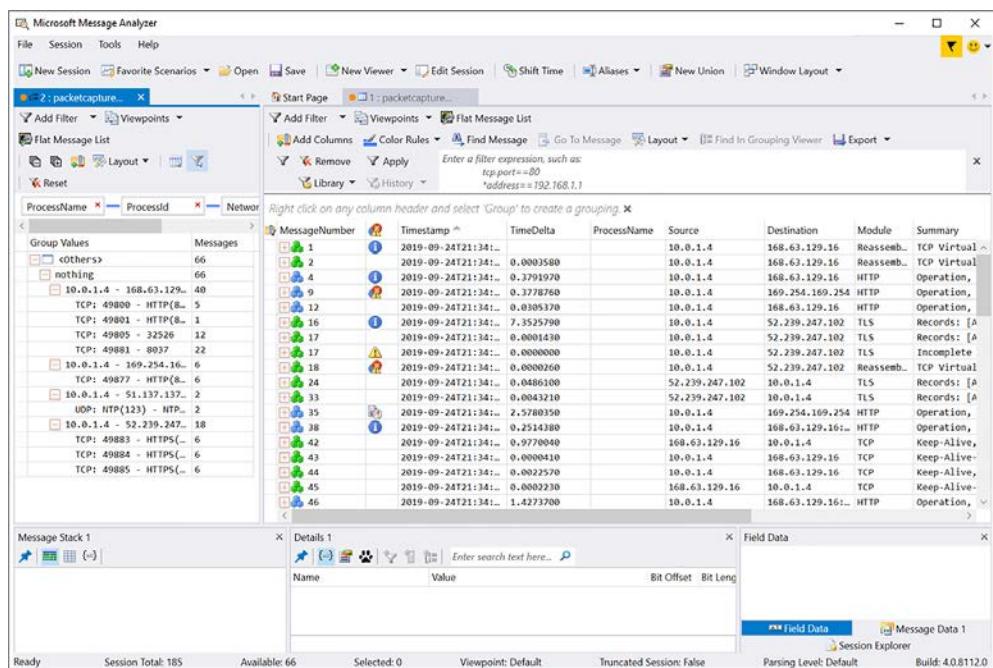


Figura 12.5 Una captura de paquetes de red visualizada en Message Analyzer de Microsoft.

Cada paquete individual está disponible para inspección. Puede agrupar y filtrar por protocolo de comunicación o cliente-host. Esta profundidad de datos de red permite examinar los paquetes reales que fluyen entre nodos para solucionar problemas cuando se produce un error. Un antiguo colega me dijo una vez: "los paquetes nunca mienten". El enigma es averiguar lo que le quieren decir los paquetes.

y filtre los paquetes de red, típicamente agrupándolos mediante comunicaciones o protocolos relacionados. En la figura 12.5 se muestra un ejemplo de captura de paquetes de red.

Para habilitar Network Watcher para capturar paquetes desde y hacia las VM, primero Instale la extensión VM de Network Watcher. Como vio en la sección 12.3.2, las extensiones de VM proporcionan una manera para que la plataforma Azure llegue al interior de una VM para realizar varias tareas de administración. En el caso de la extensión de Network Watcher, examina el tráfico de red desde y hacia la VM.

Pruébelo ahora

Siga los pasos descritos a continuación para instalar los paquetes de red de captura y extensión de VM de Network Watcher:

- 1 En Azure Portal, seleccione Máquinas virtuales en el menú de la izquierda y, a continuación, seleccione la VM, como molvm.
- 2 En la categoría Configuración a la izquierda en la ventana de VM, seleccione Extensiones.
- 3 Seleccione Agregar una extensión.
- 4 En la lista de extensiones disponibles, seleccione Agente de Network Watcher para Windows y luego seleccione Crear.
- 5 Para confirmar la instalación de la extensión, seleccione Aceptar. La instalación del agente de Network Watcher puede tardar algunos minutos en instalarse en su VM.
- 6 Para volver al menú de Network Watcher en Azure Portal, seleccione Todos los servicios en la parte superior del menú de navegación de Servicios a la izquierda del portal y, a continuación, seleccione Network Watcher.
- 7 En la sección Herramientas de diagnóstico de red a la izquierda en la ventana de Network Watcher, seleccione Captura de paquetes y, a continuación, seleccione Agregar una nueva captura.
- 8 Seleccione el grupo de recursos, como azuremolchapter12, y la VM, como molvm; luego escriba un nombre para su captura de paquetes, como molcapture.

De forma predeterminada, las capturas de paquetes se almacenan en Azure Storage. También puede elegir guardar en archivo y especificar un directorio local en la VM de origen. La extensión del agente de Network Watcher luego escribe el archivo de captura de paquetes en el disco en la VM.

- 9 Si no está ya seleccionado, elija el nombre de la cuenta de almacenamiento que comienza con el nombre del grupo de recursos, como azuremolchapter12diag493. Esta cuenta de almacenamiento creada y utilizada por la extensión de diagnóstico de VM que habilitó anteriormente.
- 10 Puede especificar un tamaño máximo para cada paquete (el valor predeterminado es 0 para todo el paquete), el tamaño máximo del archivo para la sesión de captura de paquetes (el valor predeterminado es 1 GB) y el límite de tiempo para la captura de paquetes (el valor predeterminado es 5 horas). Para capturar solo el tráfico de fuentes o puertos específicos, también puede agregar un filtro para reducir el alcance de las capturas de paquetes.

- 11 Defina un límite de tiempo de 60 segundos.
- 12 Para iniciar la captura de paquetes, seleccione Aceptar.

El inicio de la captura tarda uno o dos minutos. Cuando la captura está en curso, los datos se transmiten a la cuenta de Azure Storage o al archivo local de la VM. La lista de capturas se muestra en la página del portal de Network Watcher. Si transmite los registros a Azure Storage, puede hacer que la captura vaya directamente a la cuenta de almacenamiento y descargar el archivo de captura .cap. A continuación, puede abrir la captura de paquetes en un programa de análisis, como se analizó en la sección 12.3.3. De hecho, la captura de red de ejemplo mostrada en la figura 12.5 anteriormente en este capítulo era de una captura de paquetes de Azure Network Watcher.

12.4 Laboratorio: Creación de alertas de rendimiento

Espero que el diagnóstico, las métricas de VM y las funciones de Network Watcher abordadas en este capítulo le hayan dado una idea de lo que hay disponible en Azure para ayudarle a solucionar problemas con las aplicaciones. Algunas cosas, como los diagnósticos de arranque y la extensión de diagnóstico de VM, tienen más sentido cuando las habilita y configura a medida que implementa las VM.

En este laboratorio, configurará algunas alertas de métricas para ver qué notificaciones puede recibir y cómo se verán las alertas cuando las reciba:

- 1 En Azure Portal, busque la VM que creó en los ejercicios anteriores.
- 2 En la sección Monitor de la VM, seleccione Alertas.
- 3 Elija crear una regla de alerta y, a continuación, agregue una condición para cuando el porcentaje de la CPU sea superior a un promedio del 10 % en los últimos 5 minutos. Un gráfico le mostrará cuáles son las métricas más recientes, así que ajuste el umbral si el 10 % no activó una alerta.
- 4 Agregue un grupo de acciones y asigne un nombre y un nombre corto. Para este laboratorio, establece ambos nombres como azuremol. Los grupos de acciones le permiten definir conjuntos de pasos reutilizables para realizar cuando se genera una alerta, como enviar un correo electrónico a un conjunto de usuarios o ejecutar un script de PowerShell automatizado o una Azure Logic App.
- 5 Explore los tipos de acción disponibles y, a continuación, seleccione correo electrónico/SMS/Push/de voz.
- 6 Elija cómo quiere que se le notifique, por ejemplo, por correo electrónico o mensaje de texto. Es posible que se apliquen algunos cargos del operador por las notificaciones de SMS o de voz.
- 7 Una vez creado el grupo de acciones, asigne un nombre a la alerta y luego especifique una seguridad. Esta gravedad es útil cuando tiene muchas alertas definidas para ayudarle a clasificar y priorizar lo que debe resolver primero.
- 8 Cuando esté listo, cree la regla. La regla tarda entre 10 y 15 minutos en activarse y generar las notificaciones definidas.

Este ejemplo es básico, así que piense en las alertas y notificaciones existentes que tiene para las aplicaciones y servicios y en cómo podría utilizar esta función cuando ejecute cargas de trabajo en Azure.

Parte 3

Seguro por defecto

E

n un mundo en línea en que las aplicaciones están típicamente conectadas a Internet las 24 horas del día, los 7 días de la semana, la amenaza de un ataque digital es demasiado real. Estos ataques cuestan tiempo, dinero y confianza del cliente. Una parte central del desarrollo de aplicaciones altamente redundantes y distribuidas incluye el modo para protegerlas y proteger sus datos. Azure tiene varias funciones integradas para asegurar sus datos, incluido el cifrado, control, almacén de claves digitales y copias de seguridad. En esta parte del libro, usted aprenderá a asegurar y proteger sus aplicaciones desde el principio.

Copias de seguridad, recuperación y replicación

Los siguientes capítulos presentan algunas de las características y servicios principales de Azure que le permiten crear seguridad en sus aplicaciones. Eso es probablemente demasiado subjetivo: la seguridad no debería ser una característica o consideración adicional. En su lugar, la seguridad debería estar inherentemente incorporada en el corazón y el alma de su aplicación desde el principio. En este capítulo, comenzará su viaje hacia la seguridad de Azure sobre cómo hacer una copia de seguridad y recuperar sus datos. Es posible que las copias de seguridad no parezcan un tema de seguridad común, pero piense en la seguridad como algo más que el cifrado de datos o los certificados SSL de los sitios web. ¿Qué pasa con la protección de sus datos de interrupciones, pérdida de datos y piratería? Un debate sobre copias de seguridad y replicación también es un buen tema para pasar del capítulo sobre alta disponibilidad y este capítulo.

Las copias de seguridad pueden parecer triviales, y como antiguo administrador de copias de seguridad, puedo decirle que no hay mucha emoción por los trabajos de copias de seguridad y las rotaciones. Pero las copias de seguridad oportunas que funcionan son cruciales para proteger sus aplicaciones y garantizar que en el peor de los casos, pueda restaurar sus datos de forma rápida y fiable. También puede replicar sus VM de una región Azure a otra. Esta capacidad se basa en los conceptos de alta disponibilidad que analizamos en el capítulo 7.

En este capítulo, aprenderá cómo hacer una copia de seguridad y restaurar VM y, a continuación, replicar VM automáticamente en Azure. Todas estas copias de seguridad y los puntos de restauración están cifrados para asegurar sus datos.

13.1 Azure Backup

Una de las cosas interesantes sobre Azure Backup es que es un servicio y un gran sector de almacenamiento para las copias de seguridad reales. Azure Backup puede proteger las VM en Azure, las VM o los servidores físicos locales, e incluso las VM en

otros proveedores como Amazon Web Services (AWS). Las copias de seguridad de datos se pueden almacenar en sus propias matrices de almacenamiento locales o en un almacén de recuperación Azure. La figura 13.1 muestra cómo el servicio de copias de seguridad Azure puede proteger y organizar todas sus necesidades de copias de seguridad.

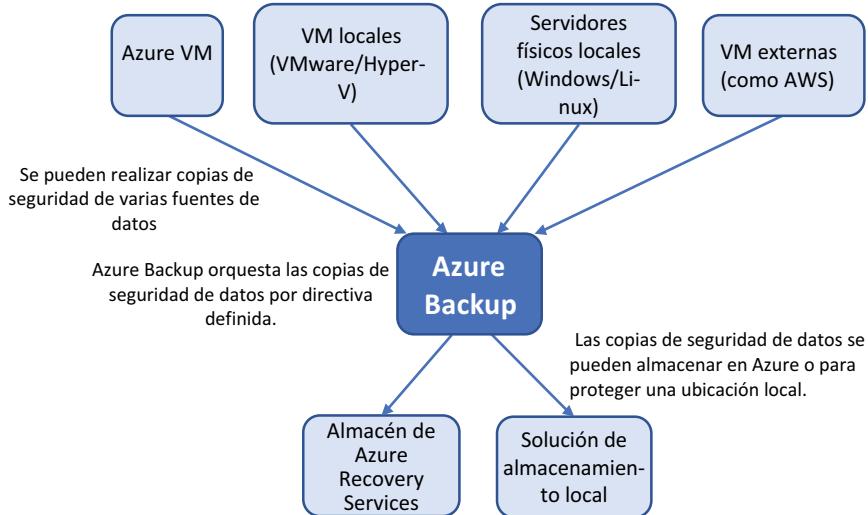


Figura 13.1 Se puede realizar una copia de seguridad de varias VM o servidores físicos, desde varios proveedores y ubicaciones, mediante el servicio de orquestación central. Azure Backup utiliza directivas definidas para realizar copias de seguridad de datos en una determinada frecuencia o programa. A continuación, estas copias de seguridad se pueden almacenar en Azure o en una solución de almacenamiento local. En todo el mundo, los datos se cifran para brindar mayor seguridad.

En su esencia, Azure Backup administra los programas de copia de seguridad y retención de datos, y organiza los trabajos de copias de seguridad y restauración. Para realizar una copia de seguridad de las VM de Azure, no tiene que instalar ningún componente de servidor ni ningún agente para instalar manualmente. Todas las operaciones de copia de seguridad y restauración están incorporadas en la plataforma Azure.

Para realizar una copia de seguridad de las VM o de los servidores físicos o las VM en otros proveedores, como AWS, instala un pequeño agente que permite la comunicación segura desde y hacia Azure. Esta comunicación segura garantiza que sus datos se cifren durante la transferencia.

Para los datos almacenados en Azure, las copias de seguridad se cifran con una clave de cifrado que usted crea. Solo usted tiene acceso a esas copias de seguridad cifradas. También puede realizar copias de seguridad de las VM cifradas (que veremos en el capítulo 14) para asegurarse de que sus copias de seguridad de datos sean seguras.

No hay ningún cargo por el flujo de tráfico de la red para hacer una copia de seguridad o restaurar los datos. Usted solo paga por cada instancia protegida y luego por mucho almacenamiento que consume en Azure. Si utiliza una ubicación de almacenamiento local, el costo de uso de Azure Backup es mínimo, ya que no hay costos de almacenamiento o de tráfico de red Azure.

13.1.1 Directivas y retención

Azure Backup utiliza un modelo de copia de seguridad incremental. Cuando usted protege una instancia, la primera operación de copia de seguridad realiza una copia de seguridad completa de los datos. Luego, cada operación de copia de seguridad realiza una copia de seguridad incremental de los datos. Cada una de estas copias de seguridad se *denomina punto de recuperación*. Las copias de seguridad incrementales son un enfoque eficaz que optimiza el uso del ancho de banda del almacenamiento y de la red. Solo los datos que han cambiado desde la copia de seguridad anterior se transfieren de forma segura a la ubicación de copia de seguridad de destino. En la figura 13.2, se detalla el modo en el que funcionan las copias de seguridad incrementales.

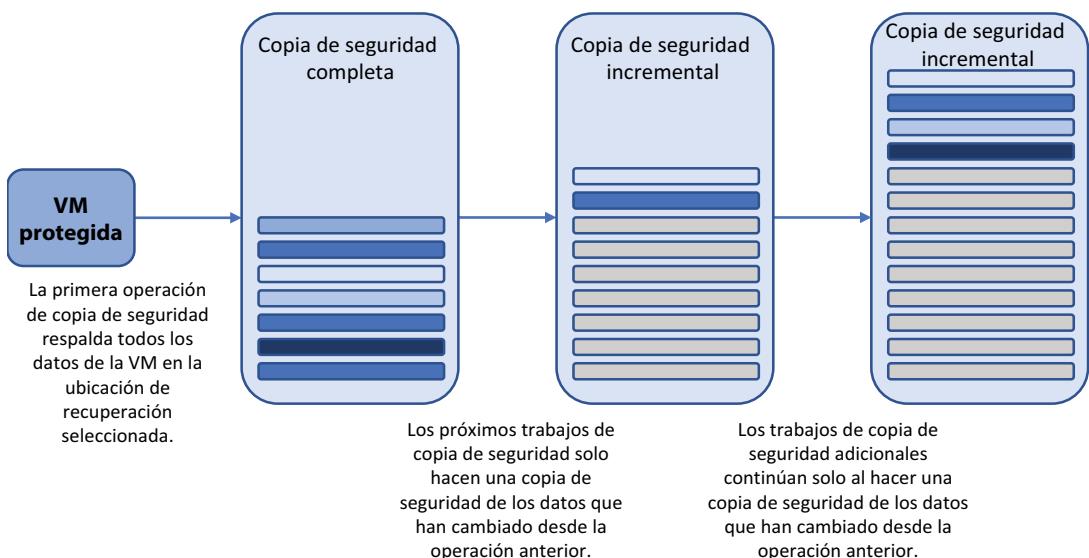


Figura 13.2 Las copias de seguridad Incrementales solo crean una copia de seguridad de los datos que han cambiado desde la operación anterior. La primera copia de seguridad siempre es una copia de seguridad completa. Cada trabajo de copia de seguridad posterior solo realiza una copia de seguridad de los datos que han cambiado desde el trabajo anterior. Controla la frecuencia de copias de seguridad completas con directivas. Este enfoque minimiza la cantidad de datos que necesita mover de forma segura a través de la red y alojar en la ubicación de almacenamiento de destino. Azure Backup mantiene las relaciones de copias de seguridad incrementales entre sí para garantizar que al restaurar datos, sean coherentes y completos.

Con Azure Backup, puede almacenar hasta 9999 puntos de recuperación para cada instancia que proteja. Para brindar contexto, si usted hiciera una copia de seguridad diaria normal, estaría establecido por más de 27 años. Y podría mantener copias de seguridad semanales por casi 200 años. ¡Creo que eso cubriría la mayoría de las situaciones de auditoría! Puede optar por retener las copias de seguridad de forma diaria, semanal, mensual o anual, que normalmente está alineado con la mayoría de las directivas de copias de seguridad existentes.

Para implementar la estrategia de copia de seguridad óptima para su carga de trabajo, debe comprender y determinar su *objetivo de punto de recuperación* (RPO) y su *objetivo de tiempo de recuperación* (RTO) aceptables.

OBJETIVO DE PUNTO DE RECUPERACIÓN

El RPO define el punto al que su copia de seguridad más reciente le permite restaurar. De forma predeterminada, Azure Backup hace una copia de seguridad diaria. A continuación, define las directivas de retención en cuanto a la cantidad de días, semanas, meses o años que desea mantener estos puntos de recuperación. Si bien el RPO se utiliza típicamente para definir la cantidad máxima de pérdida de datos aceptable, también debería considerar cuánto tiempo atrás desearía ir. En la figura 13.3, se muestra cómo el RPO define la cantidad de pérdida de datos aceptable.

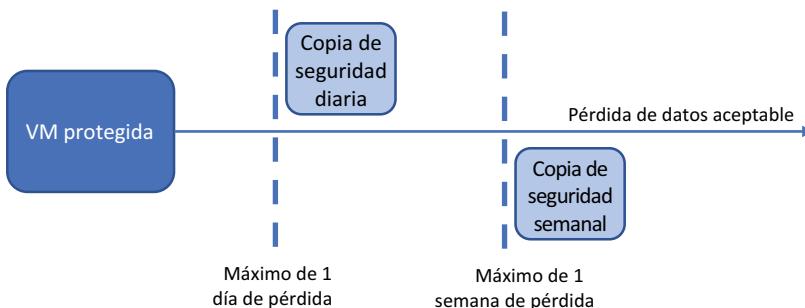


Figura 13.3 El objetivo de punto de recuperación (RPO) define la cantidad de pérdida de datos que puede mantener para una instancia protegida. Cuanto más largo sea el plazo del RPO, mayor será la pérdida de datos aceptable. Un RPO de un día significa que se podrían perder hasta 24 horas de datos, dependiendo de cuándo se produjo la pérdida de datos en relación con la última copia de seguridad. Un RPO de una semana significa hasta siete días de datos que se podrían perder.

Las interrupciones importantes y las grandes cantidades de pérdida de datos son hechos poco habituales. Más comunes son los incidentes de pequeñas pérdidas de datos o sobrescrituras. Estos incidentes a menudo no se notan ni se informan hasta después de la pérdida de datos. Aquí es donde la directiva de retención de las instancias protegidas se vuelve importante. Si tiene una directiva de retención a corto plazo, es posible que no pueda restaurar los datos desde el punto requerido. Es necesario determinar un equilibrio entre la retención de varios puntos de recuperación y los costos de almacenamiento para conservar todos esos puntos de recuperación.

Azure Storage es relativamente económico: en general menos de USD 0,02 por gigabyte de almacenamiento. Esto equivale a aproximadamente USD 2 por mes para una copia de seguridad de datos de VM de 100 GB (más un cargo por el servicio de Azure Backup). Según cuánto cambien los datos, el tamaño de los puntos de recuperación incrementales podría incrementarse rápidamente. La retención de puntos de recuperación durante semanas o meses podría costar decenas de dólares por mes por instancia protegida. Esto no es para desanimarlo, pero es importante planificar sus necesidades y ser inteligente sobre sus costos. El almacenamiento parece económico a menos de USD 0,02 por gigabyte hasta que tenga cientos de gigabytes por instancia protegida y docenas o incluso cientos de instancias para proteger.

Soy un antiguo administrador de copias de seguridad, y la capacidad de almacenamiento a menudo era un factor central cuando determinaba cuántos puntos de recuperación conservar. Esta capacidad de almacenamiento creaba en general

acuerdos con esos RPO. Si utiliza Azure Storage en lugar de una solución de almacenamiento local, no tendrá que preocuparse por la capacidad de almacenamiento disponible. ¡Puedo garantizar que hay más almacenamiento que el límite de su tarjeta de crédito!

OBJETIVO DE TIEMPO DE RECUPERACIÓN

El RTO indica la rapidez con la que puede restaurar los datos. Si decide realizar una copia de seguridad de las VM de Azure y almacenar los puntos de recuperación en una solución de almacenamiento local, tardará mucho más tiempo en restaurar esas copias de seguridad que si se alojan directamente en Azure Storage. Lo contrario sería cierto si hiciera una copia de seguridad de las VM locales o los servidores físicos en Azure Storage. En la figura 13.4, se describe el RTO.

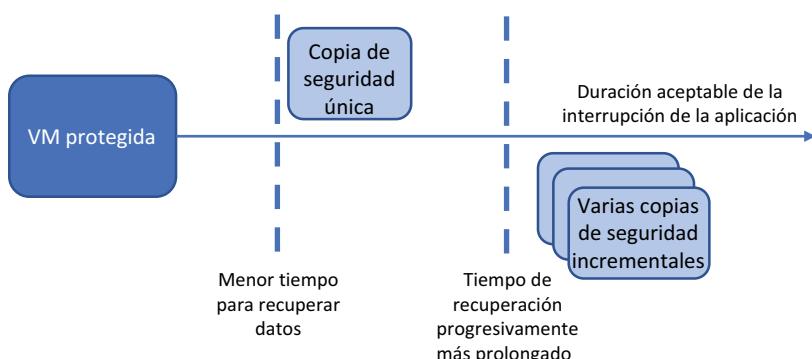


Figura 13.4 El RTO define el tiempo que es aceptable que tarde el proceso de restauración de datos y la aplicación no esté disponible. Mientras más puntos de recuperación estén involucrados en el proceso de restauración, más largo será el RTO. De manera similar, cuanto más cerca esté el almacenamiento de copias de seguridad al punto de restauración, más corto será el RTO.

En cualquier escenario, los datos de punto de recuperación deberían transferirse desde la ubicación de almacenamiento del punto de recuperación a la ubicación de restauración. Para operaciones de restauración grandes, en que es posible que necesite transferir cientos de gigabytes, el ancho de banda de la red se convierte en un verdadero cuello de botella que controla la rapidez con la que puede hacer que las aplicaciones estén disponibles nuevamente.

Lo mismo se aplica a las directivas de retención prolongadas con muchos puntos de recuperación incrementales sucesivos. La restauración de los datos puede requerir que se monten y restauren varios puntos de recuperación. Su trabajo es determinar cuánto tiempo atrás debería poder ir y cuánto tiempo puede tomarse para restaurar los datos.

Tanto el RPO como el RTO varían en función de las aplicaciones y el uso empresarial. Una aplicación que procesa los pedidos en tiempo real no puede tolerar mucha interrupción o tiempo de inactividad, por lo que es probable que el RPO y el RTO sean muy bajos. Por lo general, se utiliza una base de datos para contener los datos, por lo que normalmente se diseñan tolerancias en la aplicación en lugar de depender de los puntos de recuperación. Si se remonta a Cosmos DB, no hay nada que respaldar: la plataforma Azure realiza la replicación y protección de datos por usted. Si construyó una solución personalizada en MySQL o Microsoft SQL Server, normalmente usaría un

tipo similar de clústeres y replicación para garantizar que existan varias copias de la base de datos, por lo que la pérdida de una instancia no necesitaría que restaure desde una copia de seguridad. Las copias de seguridad son principalmente para brindar protección contra una interrupción importante o el daño de datos.

13.1.2 Programaciones de copia de seguridad

¿Cómo controla la frecuencia de sus copias de seguridad y la retención de los puntos de recuperación? En Azure Backup, estos ajustes se definen en las directivas. Usted crea estas directivas para cubrir los distintos escenarios en los que desea brindar protección y puede reutilizar las directivas para varias instancias protegidas.

Por ejemplo, una directiva de copia de seguridad puede definir que desea realizar una copia de seguridad a las 6:30 p.m. cada día. Desea mantener las copias de seguridad diarias durante seis meses y rotarlas para retener copias de seguridad semanales durante dos años. Para cumplir con la normativa, retiene copias de seguridad mensuales durante 5 años. Se retiene una copia de seguridad anual durante 10 años. Estos valores de retención pueden parecer excesivos, pero para una aplicación que implica comunicación y mensajería, a menudo es necesario retener copias de seguridad con fines regulatorios y de cumplimiento de normativas para estos largos plazos. Azure Backup proporciona la flexibilidad para definir directivas que se adapten a las diferentes cargas de trabajo de las aplicaciones y cumplir el reglamento.

Pruébelo ahora

Todas sus copias de seguridad de Azure se almacenan en un almacén de servicios de recuperación. Para crear una directiva de almacén y copia de seguridad, complete los pasos siguientes:

- 1 Abra Azure Portal y seleccione Crear un recurso en la parte superior izquierda del menú.
- 2 Busque y seleccione Copia de seguridad y Recuperación de sitios y, a continuación, elija Crear.
- 3 Cree un nombre de recurso, como `azuremolchapter13`, y luego ingrese un nombre para el almacén, como `azuremol`.
- 4 Seleccione una ubicación y, a continuación, revise y cree el almacén.
- 5 Cuando se haya creado el almacén, seleccione Grupos de recursos en el menú de la izquierda en el portal y, a continuación, elija el grupo de recursos que creó.
- 6 Seleccione el almacén de Servicios de recuperación de la lista de recursos disponibles, elija Directivas de copia de seguridad en el menú a la izquierda y, a continuación, seleccione para agregar una directiva.
- 7 Seleccione el tipo de directiva de máquina virtual de Azure y, a continuación, proporcione un nombre para su nueva directiva, como `molpolicy`. De forma predeterminada, se crea una copia de seguridad cada día.
- 8 Seleccione la zona horaria más adecuada en el menú desplegable. De forma predeterminada, Azure utiliza el horario universal coordinado (UTC).

Si lo desea, revise y ajuste las directivas de retención en diaria, semanal, mensual y anual. En la sección sobre los conceptos de copias de seguridad y calendarios de retención se detalla cómo se seleccionan estos valores. Estos valores suelen variar a medida que se crean y aplican las directivas de copia de seguridad para proteger las instancias de las máquinas virtuales.

- 9 Cuando esté listo, seleccione Crear.

La vida simple

También puede configurar copias de seguridad de VM cuando cree una VM en Azure Portal. En la página Configuración en la que configura los ajustes de red virtual o las opciones de diagnóstico y solución de problemas, puede habilitar Azure Backup. Puede elegir un almacén de Servicios de recuperación existente o crear uno, y crear o utilizar una directiva de copia de seguridad. Actualmente no puede habilitar las copias de seguridad como parte de la implementación de VM en la CLI de Azure o Azure PowerShell, pero normalmente es un comando único posterior a la implementación para hacerlo.

Me gusta planificar una estrategia de copia de seguridad, directivas de retención y programas; por eso es que estos ejercicios crearon primero el almacén y las directivas de Servicios de recuperación. Pero, si quiere crear rápidamente una VM y habilitar copias de seguridad, puede hacerlo en Azure Portal en un solo paso.

Ahora tiene una directiva de copia de seguridad, que también define las directivas de retención para varios períodos, pero no tiene nada aún para hacer una copia de seguridad. Vamos a crear una VM con Cloud Shell para que pueda crear una copia de seguridad y, en un ejercicio posterior, replicar los datos.

Pruébelo ahora

Para crear una VM de prueba para copia de seguridad y replicación, complete los pasos siguientes:

- 1 Seleccione el icono de Cloud Shell en la parte superior de Azure Portal.
- 2 Cree una VM con `az vm create`; proporcione el nombre del grupo de recursos creado en el laboratorio anterior, como `azuremolchapter13`, y, a continuación, escriba un nombre de VM, como `molvm`:

```
az vm create \
  --resource-group azuremolchapter13 \
  --name molvm \
  --image win2019datacenter \
  --admin-username azuremol \
  --admin-password P@ssw0rdMoL123
```

Se define una directiva de copia de seguridad y una VM de prueba está lista. Para ver Azure Backup en acción, vamos a aplicar su directiva de copia de seguridad a la VM.

Pruébelo ahora

Para realizar una copia de seguridad de una VM con la directiva definida, complete los pasos siguientes:

- 1 Seleccione Grupos de recursos en el menú a la izquierda del portal.
- 2 Elija el grupo de recursos y, a continuación, la VM que creó.
- 3 En Operaciones, seleccione Copia de seguridad.

- 4 Asegúrese de que su almacén de Servicios de recuperación esté seleccionado y, a continuación, elija la directiva de copia de seguridad en el menú desplegable.
- 5 Revise las opciones de programación y retención y, a continuación, active la copia de seguridad. Tarda unos segundos en aplicarse la directiva de copia de seguridad.
- 6 Cuando la directiva esté activada, vuelva a la configuración de la copia de seguridad. Los informes de estado de la VM Warning (Initial backup pending).
- 7 Para crear la primera copia de seguridad, elija el botón Hacer copia de seguridad ahora, como se muestra en la figura 13.5.

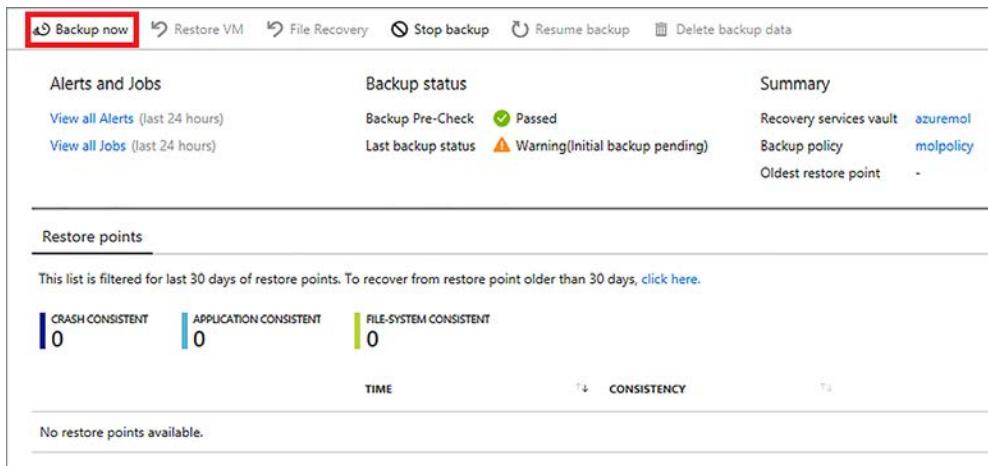


Figura 13.5 Para crear la primera copia de seguridad, seleccione el botón Hacer copia de seguridad ahora. El estado se actualiza cuando se completa y muestra el tiempo de copia de seguridad más reciente, el último punto de restauración y el punto de restauración más antiguo.

Puede tardar de 15 a 20 minutos en completar la primera operación de copia de seguridad completa. Para ver el progreso del trabajo de copia de seguridad, puede seleccionar la opción Ver todos los trabajos. No hay ninguna barra de progreso ni indicador de porcentaje, pero puede asegurarse de que el trabajo sigue ejecutándose.

¡Eso es todo lo que se necesita para hacer una copia de seguridad de VM y proteger sus datos en Azure! Siga leyendo para ver cómo puede restaurar los datos, si algo sale mal.

13.1.3 Restauración de VM

Azure Backup le permite restaurar una VM completa o realizar una restauración en el nivel de archivo. En todos mis años de trabajo, las operaciones de restauración en el nivel de archivo eran las más comunes de las dos. Este tipo de trabajo de restauración se realiza normalmente cuando los archivos se eliminan o se sobrescriben de manera accidental. Las restauraciones en el nivel de archivo suelen determinar las directivas de retención de las copias de seguridad. Cuanto más importantes sean los datos, más probable es que desee conservar las copias de seguridad durante más tiempo, en caso de que obtenga una llamada nocturna para restaurar un archivo de hace seis meses.

Una restauración de VM completa, como podría esperar, restaura toda la VM. Rara vez he realizado una restauración de VM completa para traer una VM eliminada de nuevo en línea. Un gran caso práctico para una restauración de VM completa es

proporcionar una VM de prueba, funcionalmente equivalente a la original. Puede restaurar una VM y luego probar una actualización de software u otro procedimiento de mantenimiento, lo que puede ayudarle a identificar posibles problemas y crear un plan para manejar la VM de producción real.

También es importante probar periódicamente sus copias de seguridad. No espere hasta que surja una situación en la que necesite restaurar los datos en un escenario del mundo real. Confie en Azure Backup, pero compruebe que sabe cómo y dónde restaurar los datos cuando sea necesario.

RESTAURACIÓN EN EL NIVEL DE ARCHIVO

Una restauración en el nivel de archivo es un proceso bastante genial en Azure Backup. Para darle flexibilidad en el modo y el lugar para restaurar archivos, Azure crea un script de recuperación que usted descarga y ejecuta. Este script de recuperación está protegido con una contraseña para que solo usted pueda ejecutar el proceso de recuperación. Cuando ejecute el script de recuperación, se le pedirá que introduzca la contraseña antes de poder continuar. La ventana para descargar el script de recuperación se muestra en la figura 13.6.

Al ejecutar el script de recuperación, el punto de recuperación se conecta como un sistema de archivos local en su equipo. Para las VM de Windows, se genera un script de PowerShell y se conecta un volumen local, como F:. Para las VM de Linux, el punto de recuperación se monta como un disco de datos, como /dev/sdc1 en el volumen de inicio. En ambos casos, el script de recuperación indica claramente dónde puede encontrar los archivos.

Cuando haya terminado de restaurar los archivos desde el almacén de recuperación, volverá a Azure Portal y seleccionará la opción Desmontar discos. Este proceso desconecta los discos del equipo local y los devuelve para su uso en el almacén de recuperación. No se preocupe si se olvida realizar este proceso de desmontaje en caliente cuando necesita restaurar rápidamente los archivos de una VM de producción. Azure desconecta automáticamente cualquier punto de recuperación conectado después de 12 horas.

RESTAURACIÓN COMPLETA DE VM

Una restauración completa de VM crea una VM, conecta la VM a la red virtual y conecta todos los discos duros virtuales. Probemos el proceso para una restauración completa.

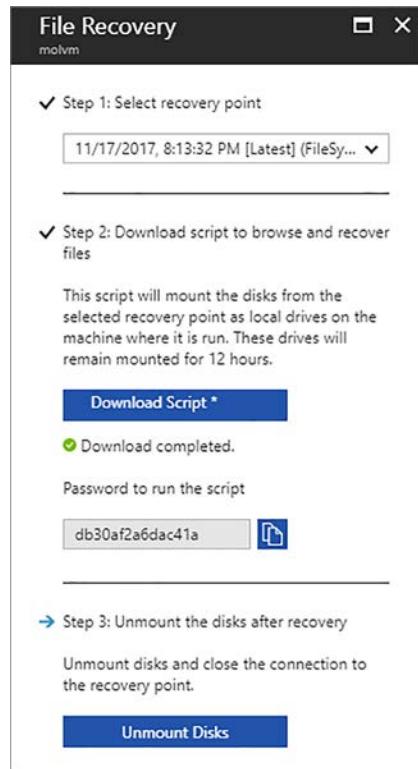


Figura 13.6 Cuando se realiza una restauración en el nivel de archivo, se elige un punto de recuperación para restaurar. Luego, se descarga un script de recuperación en su equipo. Puede ejecutar este script con solo introducir la contraseña generada. El script de recuperación monta el punto de recuperación como un volumen local en el equipo. Cuando haya restaurado los archivos que necesita, desmonta los discos del equipo, que los devuelve para su uso al almacén de recuperación.

de VM. Dado que siempre es mejor probar las actualizaciones de mantenimiento antes de realizarlas de forma real, este ejercicio de restauración es una buena práctica.

Pruébelo ahora

Para restaurar una VM completa, complete los pasos siguientes:

- 1 En el grupo de recursos, seleccione la VM de la que hizo la copia de seguridad en el ejercicio anterior.
- 2 Seleccione la opción Copia de seguridad del menú a la izquierda en la VM. La información general de la copia de seguridad debe informar que se ha creado un punto de recuperación, como se muestra en la figura 13.7. Si no, espere unos minutos y luego vuelva a este ejercicio. O simplemente lea lo que implica el proceso.

TIME	CONSISTENCY	RECOVERY TYPE
9/25/2019, 8:19:24 PM	Application Consistent	Snapshot

Figura 13.7 Cuando se completa la copia de seguridad de la VM, la página de información general muestra los datos de los últimos puntos de copia de seguridad y restauración disponibles. Para iniciar el proceso de restauración, seleccione Restaurar VM.

- 3 Seleccione el botón Restaurar VM, elija un punto de restauración de la lista y, a continuación, seleccione Aceptar.
- 4 Elija un punto de restauración y seleccione cómo restaurar la VM. Puede elegir crear una nueva VM o reemplazar una existente.

La opción predeterminada es crear una nueva VM. En esta configuración, se crea una nueva VM y se conecta a la red virtual especificada, y los discos se restauran y conectan.

También puede optar por reemplazar una VM existente. En este escenario, los discos se restauran a partir de la copia de seguridad y se conectan a la VM existente. Se conserva cualquier red virtual u otras opciones de configuración aplicadas a la VM.

- 5 Para este ejercicio, optó por restaurar a una nueva VM. Proporcione un nombre para la VM restaurada, como restoredvm, y, a continuación, revise la

configuración de la red virtual y del almacenamiento. En la producción, normalmente conecta la VM restaurada a una red virtual aislada para que no afecte al tráfico de producción.

6 Elija Aceptar y luego Restaurar.

Tarda unos minutos en conectar el punto de recuperación y crear una VM restaurada con los discos anteriores conectados. En este punto, puede conectarse a la VM restaurada para probar las actualizaciones de software o restaurar grandes cantidades de datos según sea necesario.

También puede hacer una copia de seguridad de una aplicación web, así que este enfoque no es solo para las VM. El proceso es un poco diferente, pero los conceptos son los mismos. Migrar el modelo de la aplicación a una solución PaaS como una aplicación web no significa que se puede olvidar de los conceptos básicos de las copias de seguridad y de la retención de datos.

13.2 Azure Site Recovery

¿Recuerda cuando analizamos Cosmos dB, y aprendió que con el clic de un botón sus datos se replican a una región Azure completamente diferente para brindar redundancia y tolerancia a errores? ¡También puede hacerlo con VM enteras! Azure Site Recovery es un servicio potente que puede hacer mucho más que solo replicar VM en una región diferente. En la figura 13.8 se describe cómo funciona Azure Site Recovery para organizar cargas de trabajo entre ubicaciones.

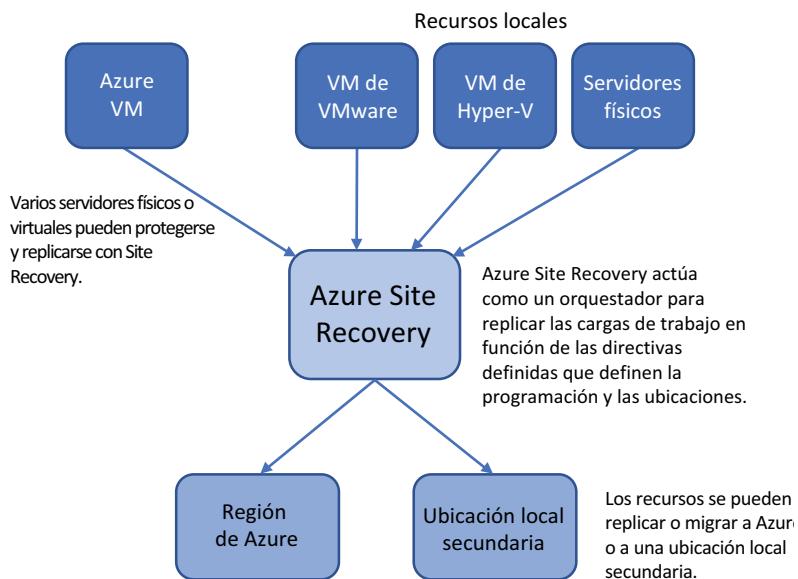


Figura 13.8 Azure Site Recovery organiza la replicación y migración de recursos físicos o virtuales en otra ubicación. Tanto las ubicaciones locales como las de Azure pueden servir como puntos de origen y destino para protección, replicación o migración.

Un aspecto importante es que Azure Site Recovery es para más que solo VM Azure: Site Recovery se puede utilizar para replicar los VMware o Hyper-V VM locales en Azure

para la recuperación ante desastres (DR) o como parte de una migración a Azure. También puede utilizar Azure Site Recovery puramente como organizador para replicar VM locales de una ubicación a una ubicación local secundaria.

De la misma manera que Azure Backup no implica que "solo funciona con Azure", Azure Site Recovery no implica que "solo replica VM de Azure". Tanto Azure Backup como Azure Site Recovery se pueden utilizar como soluciones híbridas para copias de seguridad y recuperación ante desastres. Estos servicios Azure se pueden utilizar para proteger todas sus cargas de trabajo, tanto en entornos locales como en Azure. Luego, se puede generar una única estructura para el cumplimiento y la validación para garantizar que todas las cargas de trabajo que cree que están protegidas estén realmente seguras contra la pérdida de datos.

¿Por qué utilizaría Azure Site Recovery? Dos razones principales son las más comunes: la replicación y la migración.

La replicación lo protege de una interrupción completa de la región Azure. Se necesitaría un evento catastrófico para que toda una región se desconectara, pero cuando trabaja en TI, sabe que todo es posible. Incluso los conjuntos de disponibilidad y las zonas de disponibilidad, que analizamos en el capítulo 7, normalmente solo lo protegen de una interrupción más pequeña dentro de una región Azure. Si toda la región queda inactiva, la aplicación quedará inactiva. Con Site Recovery, todo su entorno de aplicaciones, incluidos los recursos de red virtual, se replica en una región Azure secundaria. Con el clic de un botón, esa ubicación secundaria puede ser puesta en línea y activarse. Luego, el tráfico puede dirigirse a esta ubicación secundaria y comenzar a atender a sus clientes. En la figura 13.9, se muestra información general de alto nivel sobre cómo Azure Site Recovery protege su entorno.

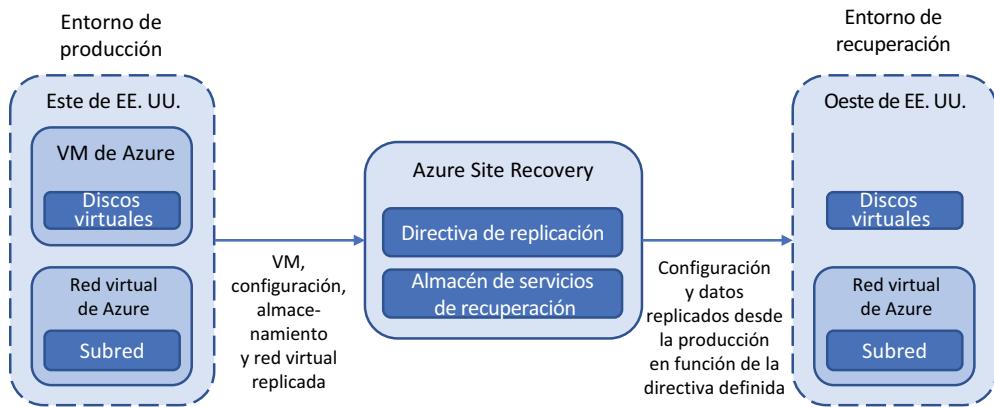


Figura 13.9 Azure Site Recovery replica la configuración, los datos y las redes virtuales desde el entorno de producción a un entorno de recuperación. Las VM no se crean en el entorno de recuperación hasta que se inicie una conmutación por error. Solo se replican los datos.

La VM es solo metadatos que definen el tamaño de la VM, los discos que están conectados y los recursos de red a los que se conecta la VM. Estos metadatos se replican, lo que permite que las VM se creen rápidamente cuando se inicia una conmutación

por error. Los discos virtuales se replican en el entorno de recuperación y se adjuntan cuando se crea una VM de recuperación durante un evento de conmutación por error.

Para la replicación de Azure a Azure, no hay un programa de replicación definido. Los discos se replican en tiempo casi real. Cuando cambian los datos de los discos virtuales de origen, se replican en el entorno de recuperación. Para las cargas de trabajo híbridas, en las que se protegen las VMware o las VM HyperV locales, se definen las directivas que controlan el programa de replicación.

Si nos centramos en la replicación de Azure a Azure, ¿cómo se replican los datos en tiempo casi real? Se crea una memoria caché de cuentas de almacenamiento en la ubicación del entorno de producción, como se muestra en la figura 13.10. Los cambios escritos en los discos virtuales de producción se replican inmediatamente en esta memoria caché de cuentas de almacenamiento. Luego, la memoria caché de la cuenta de almacenamiento se replica en el entorno de recuperación. Esta memoria caché de cuentas de almacenamiento actúa como búfer para que cualquier retraso de replicación a la ubicación de recuperación lejana no afecte el rendimiento de la carga de trabajo de producción.

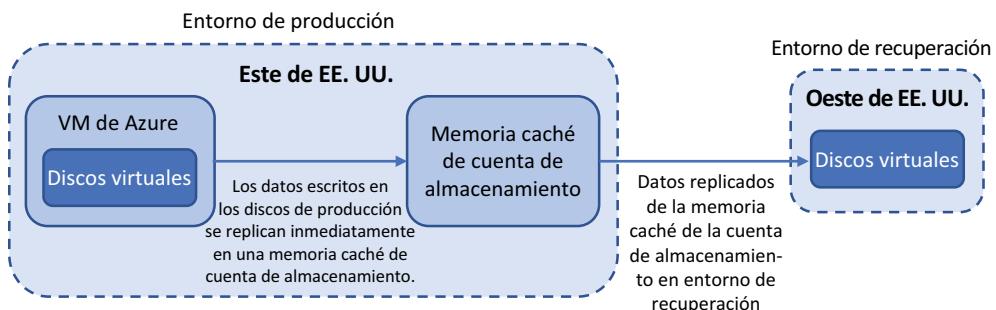


Figura 13.10 Los cambios en los discos de producción se replican inmediatamente en una memoria caché de cuentas de almacenamiento. Esta memoria caché de cuentas de almacenamiento previene el impacto del rendimiento en las cargas de trabajo de producción mientras espera replicar los cambios en la ubicación de recuperación remota. A continuación, los cambios de la memoria caché de la cuenta de almacenamiento se replican en el punto de recuperación remota para mantener la coherencia de los datos.

El proceso de configuración de Site Recovery para la replicación de Azure a Azure es sencillo, pero lleva algún tiempo crear todos los recursos replicados necesarios y completar la replicación de datos inicial. En el laboratorio de fin de capítulo, configurará esta replicación de Azure a Azure.

¿Qué puede hacer con las VM replicadas en una ubicación secundaria con Azure Site Recovery? ¡En su mayor parte, cruce los dedos y espere no necesitarlas! Pero hay un par de escenarios en los que las necesitaría.

El primero debería ser obvio: una interrupción importante. Si una región Azure no está totalmente disponible, por ejemplo, debido a un desastre natural en el área, puede iniciar una conmutación por error de sus recursos. Esta conmutación por error le indica a Azure Site Recovery que cree VM en la ubicación de recuperación según los metadatos de VM replicados y, a continuación, conecte los discos duros virtuales y las conexiones de red apropiados. También puede ser proactivo aquí: si se prevé que un desastre natural llegará a una región Azure, puede iniciar una conmutación por error *antes* de

que ocurra el evento. Este enfoque le permite decidir cuándo debe tener algún tiempo de inactividad potencial a medida que los recursos realizan una conmutación por error en la ubicación secundaria, normalmente fuera del horario hábil principal. Cuando haya pasado el evento de pronóstico en la región Azure principal, puede realizar la conmutación por recuperación de los recursos y continuar ejecutándose normalmente.

El segundo escenario en el que puede realizar la conmutación por error es para probar que el proceso funciona. De la misma manera en que las copias de seguridad se deben probar periódicamente, debe probar un plan de replicación y de conmutación por error. Sería bastante vergonzoso y estresante descubrir esto cuando necesita poner una ubicación secundaria en línea, hay alguna configuración errónea en las redes virtuales, o una de las aplicaciones no realiza la conmutación por error fácilmente. Prácticamente, Azure ofrece una opción específica para probar la conmutación por error. Una red virtual de Azure aislada se utiliza de manera normal en la ubicación secundaria y las cargas de trabajo de producción continúan ejecutándose normalmente en la ubicación primaria. Si utiliza Azure Site Recovery, asegúrese de probar periódicamente el proceso de conmutación por error.

13.3 Laboratorio: Configuración de una VM para Site Recovery

Hay varios requisitos previos para configurar la replicación de VMware o Hyper-V locales con Azure Site Recovery. Es una gran característica, tanto para propósitos de recuperación ante desastres como para migrar VM a Azure, ¡pero lleva mucho más que su descanso para almorzar! Si desea obtener más información sobre esos escenarios, diríjase a <http://mng.bz/x71V>.

Configuremos la replicación de Azure a Azure con la VM de prueba que se creó y de la que se hizo la copia de seguridad anteriormente:

- 1 En Azure Portal, seleccione Grupos de recursos en el menú de la izquierda.
- 2 Seleccione el grupo de recursos que utilizó en los ejercicios anteriores, como azuremolchapter13.
- 3 Seleccione la VM que creó en los ejercicios anteriores, como molvm.
- 4 Seleccione Recuperación ante desastres del menú de la izquierda en la ventana de VM.
- 5 En la configuración avanzada, fíjese en la configuración predeterminada que utiliza Azure Site Recovery para crear un grupo de recursos y una red virtual en la ubicación de destino. Se crea una memoria caché de cuentas de almacenamiento para replicar desde los discos virtuales de origen y se crea una directiva y un almacén de Recovery Services para controlar el proceso de replicación.
- 6 No debe cambiar nada aquí, aunque si utiliza Site Recovery en la producción y tiene varias VM para proteger, deberá revisar el modo en el que se asignan las VM a las redes virtuales replicadas existentes y a las subredes. Para este laboratorio, revise y habilite la replicación con los valores predeterminados.

Ahora, vuelva al trabajo. ¡En serio! Tarda un tiempo en configurar todos los recursos replicados y completar la sincronización de datos inicial. ¡No se quede esperando a menos que su jefe esté de acuerdo con que se tome un largo almuerzo hoy!

Protección de las copias de seguridad contra la eliminación

Espero que, como mejor práctica, haya eliminado los grupos de recursos y sus recursos al final de cada capítulo para mantener sus créditos de Azure gratuitos disponibles para su uso en el resto del libro.

Si tiene VM protegidas con Azure Backup o Site Recovery, no puede eliminar el almacén de Recovery Services o el grupo de recursos de la VM. La plataforma Azure sabe que tiene datos activos que tienen una copia de seguridad o están replicados e impide que se eliminen esos recursos.

Para eliminar las VM protegidas, primero deshabilite los trabajos de copia de seguridad activos o las VM replicadas. Cuando lo haga, puede elegir retener los datos protegidos o eliminarlos. Para los ejercicios de laboratorio de este capítulo, elija eliminar los puntos de restauración. Como una característica de seguridad, Azure elimina de manera automática estos puntos de restauración y le permite deshacerlos durante 14 días. No hay nada que pueda configurar aquí, y no puede forzar la eliminación de estos puntos de restauración borrados. No lo recomiendo, pero también puede desactivar la función de eliminación temporal de un almacén de Recovery Services seleccionando las propiedades del almacén en Azure Portal.

La buena noticia es que el resto del grupo de recursos puede ser eliminado, y no se paga por estos puntos de restauración de eliminación temporal. Cuando el período de eliminación temporal de 14 días termine, es posible eliminar de forma normal el almacén de Recovery Services. El objetivo aquí es protegerlo de la eliminación accidental, o malintencionada, de los puntos de restauración y darle tiempo para darse cuenta de que son realmente necesarios y recuperarlos.

14

Cifrado de datos

La seguridad de sus datos es importante. Más específicamente, la seguridad de los datos de sus clientes es fundamental. Apenas pasó una semana sin que leyéramos en las noticias que una empresa importante descubrió una filtración de datos. A menudo, estos incidentes son provocados por una falta de seguridad, una configuración errónea o un descuido simple. En esta era digital, es demasiado fácil para los atacantes automatizar sus intentos de obtener acceso a sus datos. El tiempo para recuperarse de un incidente de seguridad en un nivel de aplicación puede ser nada comparado con el tiempo que tarda el negocio e recuperar la confianza de sus *clientes* si se expusieron sus datos.

Azure incluye características de cifrado que dificultan la afirmación de que no tiene el tiempo ni la experiencia para proteger sus datos. En este capítulo, veremos cómo cifrar los datos almacenados en Azure Storage, en los discos administrados o en la VM completa. Se han escrito libros enteros sobre cifrado de datos, y este capítulo no profundiza sobre los métodos y consideraciones de cifrado. En su lugar, verá cómo habilitar algunas de las características y servicios principales de Azure para proteger sus datos durante el ciclo de vida de la aplicación.

14.1 ¿Qué es el cifrado de datos?

Cuando usted compra algo en línea, ¿comprueba que hay un pequeño ícono de candado en la barra de direcciones para indicar que el sitio web utiliza HTTPS? ¿Por qué es malo enviar sus información de crédito a través de una conexión HTTP normal y no segura? Cada bit de datos de un paquete de red que fluye entre dispositivos podría ser potencialmente controlado y examinado. En la figura 14.1, se muestra cómo comprar en línea sin una conexión HTTPS podría ser perjudicial para el extracto de su tarjeta de crédito.

No hay excusa para que los servidores web utilicen conexiones no seguras. Cada aplicación web que crea en Azure automáticamente tiene un certificado SSL comodín aplicado.

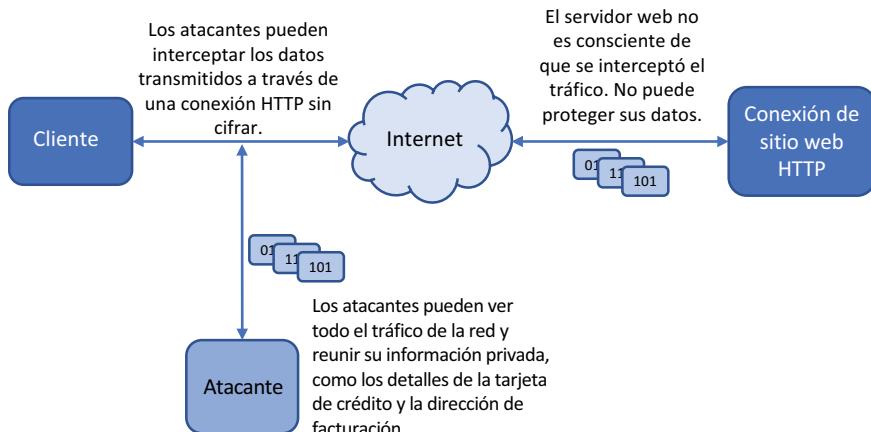


Figura 14.1 En este ejemplo básico, un atacante podría interceptar el tráfico de red que se envía a través de una conexión HTTP sin cifrar. Dado que sus datos no están cifrados, el atacante podría unir los paquetes de red y obtener su información personal y financiera. Si en su lugar se conecta al servidor web a través de una conexión HTTPS cifrada, un atacante no puede leer el contenido de los paquetes de red ni ver los datos.

Un *certificado SSL* es un componente digital que se utiliza para proteger el servidor web y permitir que un navegador web valide la conexión. Se puede utilizar un certificado SSL comodín en un dominio completo, como *.azurewebsites.net, el dominio predeterminado para aplicaciones web. Cuando creó una aplicación web en el capítulo 3, podría haber agregado https:// to the web address and started to use encrypted communications with your web apps. ¡Eso es todo!

Los certificados SSL personalizados son relativamente económicos y fáciles de implementar. A través de proyectos como Let's Encrypt (<https://letsencrypt.org>), puede obtener un certificado gratuito y configurar automáticamente su servidor web en minutos. También puede comprar y utilizar un certificado de App Service que se integra directamente en Web Apps. Los certificados de App Service se almacenan en Azure Key Vault, que analizaremos más en el capítulo 15.

A medida que diseña y construya aplicaciones en Azure, debería implementar comunicaciones seguras siempre que sea posible. Este enfoque ayuda a proteger los datos mientras están en tránsito, pero ¿qué pasa cuando esos datos se escriben en el disco? Existe un proceso similar para discos y VM que asegura y protege sus datos en reposo. En la figura 14.2, se muestra cómo funciona el cifrado de discos y VM.

Espero que estos ejemplos simplificados de cifrado de datos en Azure lo motiven a implementar el cifrado cuando diseñe y desarrolle aplicaciones en Azure. La mayoría de los clientes esperan que sus datos estén protegidos, y muchas empresas tienen mandatos reglamentarios y de cumplimiento de normativas que requieren cifrado. No piense solo en las potenciales multas para el negocio por una filtración de datos, ni la pérdida de confianza del cliente. Considere el riesgo de que los datos personales y financieros de los clientes queden expuestos y cómo esa exposición podría

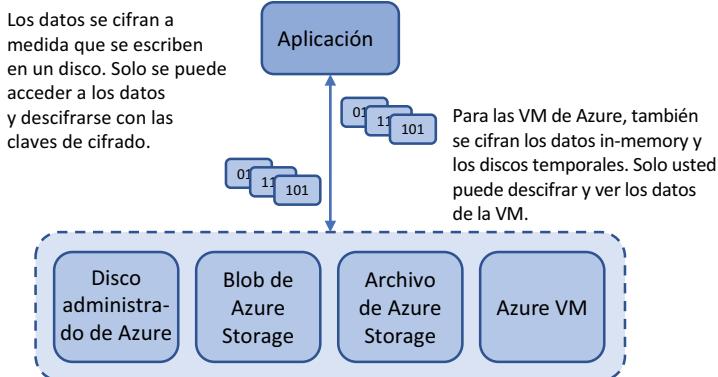


Figura 14.2 Cuando cifre los datos, solo usted podrá descifrar y ver el contenido. Si un atacante tuviera acceso a un disco virtual o a archivos individuales, no sería capaz de descifrar el contenido. Los métodos de cifrado pueden combinarse: los clientes pueden conectarse a su web a través de HTTPS, puede forzar el tráfico a cuentas de almacenamiento a través de HTTPS, y luego puede cifrar los datos que se escriben en el disco.

afectar a su vida cotidiana. Probablemente no le guste la idea de que sus propios datos estén expuestos, entonces, haga todo lo que pueda para proteger los datos de sus clientes.

14.2 Cifrado en reposo

Si el cifrado de datos es tan importante, ¿cómo lo utiliza en Azure? ¡solo siga haciendo lo que ya ha aprendido en este libro! Justo al principio, mencioné que todas las VM deberían usar discos administrados, ¿verdad? Hay muchas buenas razones para ello, una de las cuales es la seguridad. Un disco administrado se cifra automáticamente. No tiene nada para configurar y no hay ningún impacto en el rendimiento cuando está habilitado. No puede excluirse aquí: sus datos se cifran automáticamente en reposo con discos administrados.

¿Qué significa que los datos se *cifren en reposo*? Cuando utiliza discos administrados, sus datos se cifran cuando se escriben en el almacenamiento subyacente de Azure. Los datos que residen en los discos temporales, o los datos que existen in-memory de la VM, no están cifrados. Solo una vez que los datos del SO o del disco de datos *reposan* en el disco físico subyacente, se cifran. En la figura 14.3, se muestra cómo se cifran los datos a medida que se escriben en un disco administrado.



Figura 14.3 Los datos se cifran a medida que se escriben en un disco administrado. Los datos in-memory de la VM, o los datos de los discos temporales locales de la VM, no se cifran a menos que se habilite toda la VM para el cifrado, lo cual veremos más adelante en la sección 14.4.2. El cifrado automático de los datos escritos en los discos administrados no genera sobrecarga de la VM. La plataforma Azure realiza la operación de cifrado en el almacenamiento subyacente. La VM no necesita controlar ningún proceso de cifrado/descifrado.

Este cifrado en reposo para discos administrados significa que no hay impacto en el rendimiento en las VM. La VM no necesita realizar ningún procesamiento adicional para el cifrado y descifrado de los datos, por lo que se puede utilizar toda la potencia de CPU disponible para ejecutar aplicaciones. En los escenarios típicos de cifrado de VM, la VM utiliza cierta cantidad de potencia de proceso para procesar y administrar el cifrado de datos. La compensación del cifrado automático de discos administrados es que solo se protegen el sistema operativo y los discos de datos. Potencialmente, se podrían exponer otros datos de disco in-memory o temporales en la VM.

Microsoft administra las claves de cifrado digital dentro de la plataforma Azure con el cifrado automático de discos administrados. Esto crea otro intercambio en el que puede cifrar automáticamente sus datos sin necesidad de crear, administrar, rotar o revocar claves, pero tiene que confiar en Microsoft para proteger esas claves.

14.3 Storage Service Encryption

El cifrado automático de discos administrados es genial, pero ¿qué ocurre si utiliza Azure Storage para almacenamiento de blobs o archivos? Azure Storage Service Encryption (SSE) le permite cifrar los datos en el nivel de la cuenta de almacenamiento. Los datos se cifran a medida que se escriben en la cuenta. Nuevamente, Microsoft controla las claves de cifrado, por lo que no se requiere configuración ni sobrecarga de administración. La plataforma Azure extrae la generación y administración de claves para usted. Si lo prefiere, puede crear y utilizar sus propias claves de cifrado, con un poco de sobrecarga adicional de administración. Al igual que el cifrado automático del disco administrado en reposo, el cifrado del almacenamiento de Azure se activa automáticamente al crear una cuenta.

El objetivo tanto con el cifrado automático de discos administrados como con SSE es facilitarle lo más posible el cifrado de datos para que pueda dedicar más tiempo a diseñar, desarrollar y ejecutar sus aplicaciones. En la figura 14.4, se muestra cómo SSE protege sus datos y también puede forzar las comunicaciones seguras cuando los datos están en tránsito.

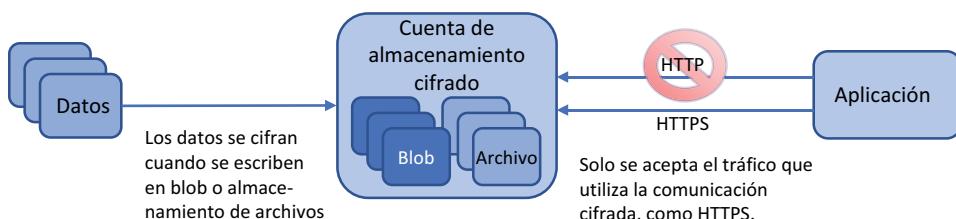


Figura 14.4 Cuando se habilita SSE, los blobs y los archivos de Azure se cifran a medida que los datos se escriben en disco. Las tablas y colas de Azure no están cifradas. Para obtener mayor seguridad de datos, puede forzar que todas las comunicaciones con una cuenta de almacenamiento utilicen protocolos de comunicación seguros, como HTTPS. Esto protege los datos en tránsito hasta el momento en que se cifran en el disco.

Forzado del tráfico de almacenamiento para utilizar transferencias seguras

Junto con la habilitación de SSE, puede forzar todas las transferencias y solicitudes de almacenamiento para utilizar un método de comunicación seguro. Esta configuración obliga a todas las llamadas de API REST a utilizar HTTPS, y todas las conexiones de archivos Azure que no habiliten el cifrado, como las versiones anteriores del protocolo SMB, se eliminarán.

(continuación)

Los SDK de Azure, como los ejemplos de Python que vimos en el capítulo 4, pueden usar conexiones cifradas. Los documentos de referencia para cada SDK específico del idioma proporcionan orientación sobre cómo implementar comunicaciones seguras.

El uso de comunicaciones seguras debe incorporarse a las aplicaciones desde el principio. Si algunos componentes no se configuraron correctamente desde el inicio, esto puede causar problemas para habilitar comunicaciones seguras en una aplicación existente. Por lo menos, pruebe las comunicaciones seguras para una aplicación existente en un entorno de desarrollo primero.

Pruébelo ahora

Para crear una cuenta de almacenamiento y habilitar el cifrado y las comunicaciones seguras, complete los pasos siguientes:

- 1 Abra el portal Azure y seleccione el icono de Cloud Shell en el menú superior.
- 2 Cree un grupo de recursos; proporcione un nombre, como azuremolchapter14; y proporcione una ubicación, como eastus:

```
az group create --name azuremolchapter14 --location eastus
```

- 3 Cree una cuenta de almacenamiento con `az storage account create`. Proporcione un nombre único, como azuremolstorage, e ingrese el grupo de recursos que creó en el paso 2. Especifique un tipo de cuenta de almacenamiento, como Standard_LRS para almacenamiento redundante local. Para forzar las comunicaciones seguras, establezca `--https-only`.

```
az storage account create \
--name azuremolstorage \
--resource-group azuremolchapter14 \
--sku standard_lrs \
--https-only true
```

- 4 Compruebe que la cuenta de almacenamiento está cifrada y habilitada para comunicaciones seguras consultando por `enableHttpsTrafficOnly` y los parámetros de cifrado:

```
az storage account show \
--name azuremolstorage \
--resource-group azuremolchapter14 \
--query [enableHttpsTrafficOnly,encryption]
```

La salida es similar a lo siguiente:

```
[  
  true,  
  {  
    "keySource": "Microsoft.Storage",  
    "useTLS": true  
  }  
]
```

```
    "keyVaultProperties": null,
    "services": [
        "blob": {
            "enabled": true,
            "lastEnabledTime": "2019-09-27T03:33:17.441971+00:00"
        },
        "file": {
            "enabled": true,
            "lastEnabledTime": "2019-09-27T03:33:17.441971+00:00"
        },
        "queue": null,
        "table": null
    }
]
```

14.4 Cifrado de VM

El cifrado automático de Azure Managed Disks ayuda a proporcionar un nivel de seguridad de VM. Para un enfoque integral de la seguridad de datos de VM, puede cifrar la VM. Este proceso implica más que cifrar los discos duros virtuales subyacentes. El disco del SO y todos los discos de datos conectados, junto con el disco temporal, están cifrados. La memoria de VM también está cifrada para reducir aún más el área de ataque. Las claves digitales se usan para cifrar las VM.

Una de las ventajas de cifrar toda la VM es que usted administra las claves de cifrado. Estas claves de cifrado se almacenan de forma segura en Azure Key Vault, y puede elegir entre usar claves generadas por software o hardware. Usted controla estas claves, de modo que puede definir el acceso a ellas y utilizar controles de acceso basados en roles y realizar auditorías para hacer un seguimiento del uso. También puede rotar las claves de cifrado en un horario definido, como cambiar su contraseña cada 60 o 90 días. Estas tareas de administración y controles adicionales para las claves de cifrado agregan un poco de sobrecarga de administración, pero proporcionan la máxima flexibilidad para proteger sus datos, y pueden requerirse para ciertos fines normativos. Veamos un poco más sobre Azure Key Vault.

14.4.1 Almacenamiento de claves de cifrado en Azure Key Vault

Pasaremos el capítulo 15 con Azure Key Vault, pero quiero mostrarle el poder del cifrado de datos y el cifrado de VM primero. Como información general rápida, Azure Key Vault es un almacén digital que le permite almacenar de forma segura claves de cifrado, certificados SSL y secretos como contraseñas. Para redundancia, los almacenes de claves se replican en regiones Azure. Esta replicación protege sus claves y secretos, y garantiza que siempre estén disponibles para su uso.

Solo usted tiene acceso a sus almacenes de claves. Genera y almacena objetos en almacenes de claves y, a continuación, define quién tiene acceso a estos almacenes. Microsoft administra el servicio de Key Vault subyacente, pero no tiene acceso al contenido de los almacenes. Este límite de seguridad significa que cuando usted cifra sus datos en Azure, usted es el único que puede descifrarlos y verlos.

Pruébelo ahora

Para crear un almacén de claves y una clave de cifrado, complete los siguientes pasos:

- 1 Abra el Azure Portal y seleccione el icono de Cloud Shell en el menú superior.
- 2 Cree un almacén de claves con el comando `az keyvault create`; especifique el grupo de recursos que creó en el ejercicio anterior, como `azuremolchapter14`; y, a continuación, proporcione un nombre único para el almacén de claves, como `azuremolkeyvault`:

```
az keyvault create \
--resource-group azuremolchapter14 \
--name azuremolkeyvault \
--enabled-for-disk-encryption
```

Vamos a hacer una pausa y pensar por qué se agrega un parámetro para `--enabled-for-disk-encryption`. Al cifrar una VM, la plataforma Azure necesita poder iniciar y descifrar la VM para que pueda ejecutarse. La plataforma Azure no tiene ningún permiso para acceder a esos datos y Microsoft no tiene acceso para ver y usar esas claves de cifrado para otra cosa que no sea iniciar una VM. Cuando se habilita un almacén de claves para el cifrado de discos, se conceden permisos para que Azure acceda al almacén de claves y utilice la clave de cifrado asociada a una VM.

Nuevamente, Microsoft no tiene acceso a estas claves ni a sus datos, solo la posibilidad de iniciar su VM cifrada. Es bastante difícil hacer mucho con una VM cifrada cuando no puede arrancar. En la figura 14.5, se muestra cómo la plataforma Azure utiliza la clave de cifrado para iniciar una VM cifrada.

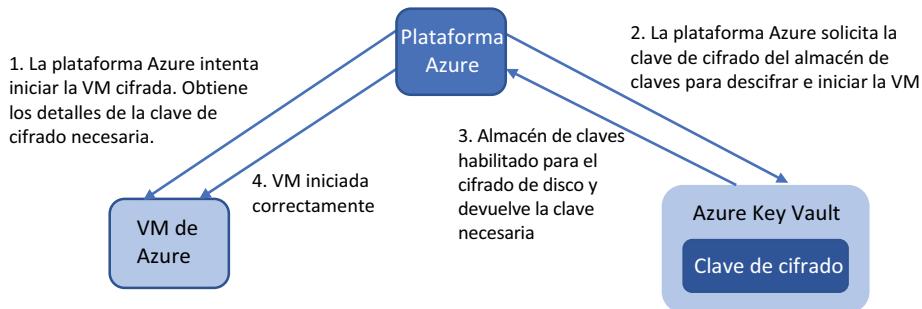


Figura 14.5 Cuando se habilita un almacén de claves para el cifrado de discos, concede permiso para que la plataforma Azure solicite y utilice la clave de cifrado para iniciar correctamente una VM cifrada.

Las claves se pueden crear y almacenar en software, o pueden ser almacenadas en los módulos de seguridad de hardware (HSM) para una mayor seguridad. Para muchos propósitos, las claves de software funcionan muy bien, aunque puede tener mandatos de seguridad que requieran el uso de HSM. Analizaremos este tema un poco más en el capítulo 15.

- 3 Para crear una clave, especifique el almacén que creó en el paso 2, como azuremolkeyvault, y, a continuación, proporcione un nombre de clave, como azuremolencryptionkey:

```
az keyvault key create \
--vault-name azuremolkeyvault \
--name azuremolencryptionkey \
--protection software
```

14.4.2 Cifrado de una VM Azure

La clave de cifrado que creó en la sección 14.4.1 puede usarse para cifrar muchas VM, si lo desea. Este enfoque minimiza la sobrecarga de la administración de claves y, si utiliza conjuntos de escala de máquinas virtuales, le permite escalar automáticamente la cantidad de instancias de VM sin necesidad de generar claves de cifrado cada vez. La alternativa es que cada VM tiene su propia clave de cifrado, que agrega complejidad, pero ofrece una capa de seguridad para sus VM. Si se tiene la misma clave de cifrado utilizada para las máquinas virtuales de aplicaciones de back-end y para las máquinas virtuales de bases de datos, por ejemplo, un atacante teórico con esa clave podría acceder a los datos de ambos conjuntos de máquinas virtuales. Si se utilizan claves diferentes, el número de máquinas virtuales potencialmente comprometidas es menor. En el laboratorio de fin de capítulo, cifrará una sola VM, aunque el mismo proceso puede funcionar con un conjunto de escalas que tenga varias VM pero que solo utilice una clave. Especialmente cuando trabaja con aplicaciones de escalado automático más grandes, asegúrese de diseñar y desarrollar características de seguridad.

Cuando se cifra una VM, se instala una extensión Azure VM. La extensión controla el cifrado del disco del SO, del disco temporal, de los discos de datos conectados y de los datos in-memory, como se muestra en la figura 14.6. Para las VM de Windows, se utiliza el mecanismo de cifrado de BitLocker. Para las VM de Linux, se utiliza dm-crypt para procesar el cifrado. La extensión VM puede informar el estado del cifrado y descifrar la VM como se desee.

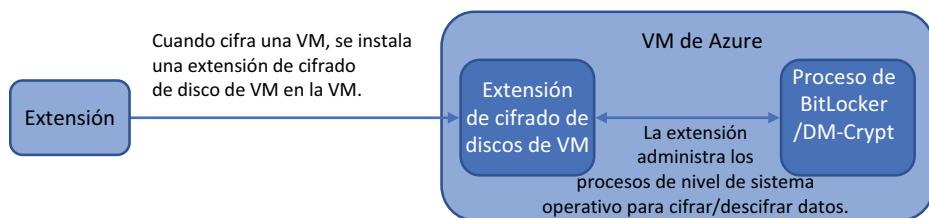


Figura 14.6 Cuando cifra una VM, se instala una extensión de cifrado de disco Azure. Esta extensión administra el uso de BitLocker en VM de Windows o dm-crypt en VM de Linux, para realizar el cifrado de datos en su VM. La extensión también se utiliza cuando se consulta el estado del cifrado de una VM.

Dado que la extensión de cifrado de disco de VM se basa en BitLocker o dm-crypt, existen algunas limitaciones sobre el uso del cifrado de VM. La mayoría de las imágenes de Azure Marketplace admiten el cifrado del disco, aunque existen algunas restricciones en los tamaños de VM que admiten cifrado o cifrado de uso compartido

de archivos de red conectada como archivos Azure. Para obtener la información más completa sobre las limitaciones y consideraciones admitidas para el cifrado de VM, lea los documentos más recientes de Azure en <http://mng.bz/yyvd>.

Este capítulo se proporcionó una introducción rápida a las características de seguridad y cifrado de datos en Azure. El cifrado automático para discos administrados y SSE no requieren mucha configuración, por lo que no hay una barrera real que le impida utilizarlos.

14.5 Laboratorio: Cifrado de una VM

Veamos todo esto en acción cifrando una VM con la clave de cifrado almacenada en el almacén de claves:

- 1 Cree una VM. La mayoría de las imágenes de Linux en Azure Marketplace admite cifrado, al igual que las imágenes de Windows Server del servidor 2008 R2 y posteriores. Para hacerlo rápido y fácil, cree una VM de Ubuntu LTS, tal como lo hizo para la mayor parte de este libro. Como la VM requiere suficiente memoria para realizar la operación de cifrado de disco, especifique un tamaño de Standard_D2s_v3:

```
az vm create \
--resource-group azuremolchapter14 \
--name molvm \
--image ubuntults \
--size Standard_D2s_v3 \
--admin-username azuremol \
--generate-ssh-keys
```

- 2 Habilite el cifrado en la máquina virtual y proporcione el nombre de Azure Key Vault que creó en un ejercicio anterior:

```
az vm encryption enable \
--resource-group azuremolchapter14 \
--name molvm \
--disk-encryption-keyvault azuremolkeyvault \
--key-encryption-key azuremolencryptionkey
```

Tarda unos minutos en instalar la extensión de cifrado de disco de Azure VM y comenzar el proceso de cifrado de la VM.

- 3 Cuando se haya iniciado el cifrado, controle el progreso y esté listo para reiniciar la VM para completar el proceso de cifrado. Vea el estado de la siguiente manera:

```
az vm encryption show \
--resource-group azuremolchapter14 \
--name molvm \
--query 'status'
```

He aquí un ejemplo de salida de una VM en el proceso de cifrado. Al inicio, el mensaje de estado informa como

```
[{"code": "ProvisioningState/succeeded",
```

```
        "displayStatus": "Provisioning succeeded",
        "level": "Info",
        "message": "OS disk encryption started",
        "time": null
    }
]
```

Puede llevar un tiempo completar el cifrado de disco, por lo que este puede ser otro buen ejercicio de laboratorio al cual regresar en una hora más o menos, a menos que desee un largo almuerzo. Yo no soy su jefe, pero *es* aburrido mirar el mismo mensaje de estado de cifrado.

- 4 Cuando el estado de cifrado se informe como Encryption succeeded for all volumes, reinicie la VM:

```
az vm restart --resource-group azuremolchapter14 --name molvm
```

A continuación, puede comprobar el estado del cifrado de la VM nuevamente con `az vm encryption show` para confirmar que la VM se informa como Cifrada.

Recuerde sus tareas de limpieza

Estos dos últimos laboratorios de fin de capítulo no tardaron mucho en completarse, pero puede que hayan tardado un poco en terminar. No olvide volver y eliminar los recursos cuando termine con ellos.

En el capítulo 13, recuerde que necesita deshabilitar la protección de Azure Backup o Site Recovery antes de poder eliminar el grupo de recursos y el almacén de Recovery Services (después de esperar los 14 días para que caduquen los puntos de recuperación de eliminación temporal de archivos gratuitos). Asegúrese de volver y limpiar esos recursos de laboratorio antes de que empiecen a usar demasiados de sus créditos gratuitos de Azure.

Protección de la información con Azure Key Vault

Casi todas las semanas hay noticias de un incidente de ciberseguridad con una empresa importante. De la misma manera que ha utilizado varias formas de automatización para hacer crecer o replicar sus aplicaciones y datos, los atacantes automatizan sus propias acciones. Es improbable que una sola persona trate de comprometer la seguridad de sus sistemas de forma manual. Este concepto dificulta la defensa de sus sistemas 24 horas al día, 7 días a la semana y 365 días al año (está bien, o 366 días).

El capítulo 14 analizó cómo cifrar sus datos y VM. Este proceso es un gran primer paso, y hemos analizado brevemente cómo crear y utilizar las claves de cifrado almacenadas con el servicio Azure Key Vault. Los datos seguros, como claves, secretos y certificados, se almacenan de mejor manera en un almacén digital como un almacén de claves, que puede administrar, emitir y auditar de forma centralizada el uso de sus credenciales y datos críticos. A medida que sus aplicaciones y servicios requieren acceso a diferentes recursos, pueden solicitar, recuperar y utilizar automáticamente estas claves, secretos y credenciales. En este capítulo, aprenderá por qué y cómo crear un almacén de claves seguro, controlar el acceso y, a continuación, almacenar y recuperar secretos y certificados.

15.1 Protección de la información en la nube

A medida que las aplicaciones se tornan más complejas y crece el riesgo de ciberataques, la seguridad se convierte en una parte fundamental de cómo diseñar y ejecutar sus servicios. Asegurarse de minimizar el riesgo del acceso a datos no autorizados, en especial cuando ejecuta más aplicaciones orientadas a Internet, ya sea locales o en la nube, debe ser una de las principales áreas de diseño en las que enfocarse. No tiene sentido tener la mejor pizzería del mundo si los clientes no confían en usted para entregarle sus detalles de pago o información personal.

Una manera común de proporcionar seguridad para las aplicaciones y los servicios es a través de claves digitales, secretos y certificados, como se muestra en la figura 15.1. En lugar de usar un nombre de usuario y una contraseña que se deben

introducir manualmente una y otra vez, o, quizás peor, escribirlas en un archivo de configuración sin cifrar, se utiliza un almacén digital para almacenar de forma segura estas credenciales y datos. Cuando una aplicación o servicio requiere acceso, solicita la clave o secreto específico que necesita, y también se crea un registro de auditoría para rastrear cualquier posible uso indebido o incumplimiento de la seguridad.

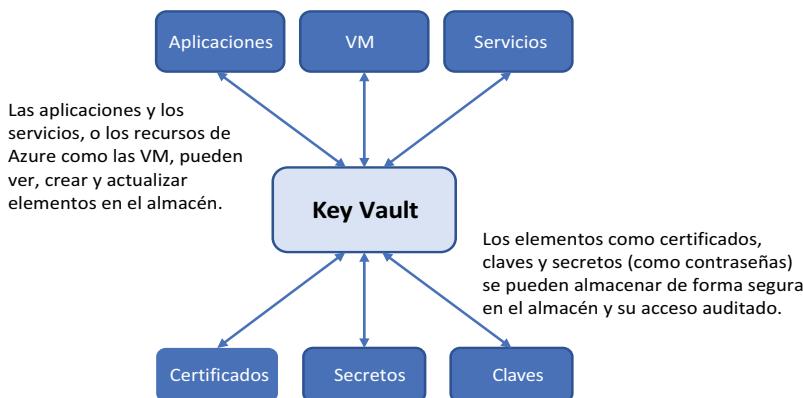


Figura 15.1 Azure Key Vault proporciona una forma segura de almacenar información digital como certificados, claves y secretos. Entonces, es posible acceder a estos elementos seguros directamente desde sus aplicaciones y servicios, o los recursos de Azure como las máquinas virtuales. Con una interacción humana mínima, puede distribuir de forma centralizada credenciales y certificados seguros en los entornos de aplicaciones.

Cuando se diseñan e implementan correctamente, estos almacenes digitales están automatizados casi por completo y seguros. Los servicios pueden solicitar un nuevo certificado digital, emitir uno que se almacene de forma segura en el almacén y usarlo para autorizarse contra otros componentes de la aplicación. Los servidores pueden configurar el software al recuperar secretos como contraseñas del almacén digital y, a continuación, instalar componentes de la aplicación, sin que las credenciales se almacenen en un archivo de configuración basado en texto. Un administrador de aplicaciones puede administrar de forma centralizada todos los secretos, claves y certificados de un servicio y actualizarlos periódicamente según sea necesario.

Azure Key Vault proporciona todas estas características de seguridad digital y le permite controlar de cerca qué usuarios y recursos pueden acceder a los datos seguros. Los almacenes de claves se pueden replicar de forma segura para redundancia y mejorar el rendimiento de las aplicaciones, e integrarse con recursos comunes de Azure, como VM, aplicaciones web y cuentas de Azure Storage.

15.1.1 Almacenes de software y módulos de seguridad de hardware

Antes de pasar de lleno a un ejemplo práctico de cómo crear y utilizar un almacén de claves, es importante comprender la forma en que se almacena la información segura en un almacén. Como se muestra en la figura 15.2, todas las claves, secretos y certificados de un almacén de claves se almacenan en un módulo de seguridad de hardware (HSM). Estos dispositivos no son únicos en Azure, sino que son dispositivos de hardware de toda la industria que proporcionan un alto nivel de seguridad para los datos almacenados en ellos.

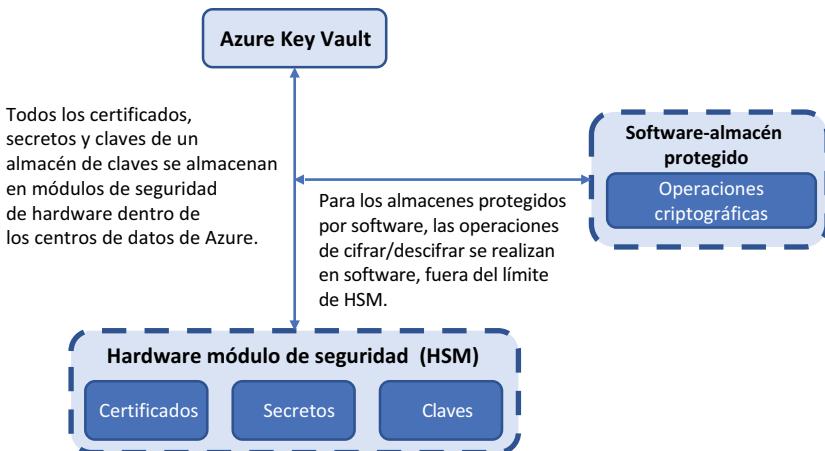


Figura 15.2 Azure Key Vault es un recurso lógico en Azure, pero todos los certificados, secretos y claves se almacenan en un HSM. Para escenarios de desarrollo o pruebas, se puede utilizar un almacén protegido por software, que luego realiza cualquier operación de criptografía (como cifrar o descifrar datos) en software y no en hardware en el HSM. Para la producción, debe utilizar un almacén protegido por HSM, donde todo el procesamiento se realiza en hardware.

En la actualidad, puede usar hay dos tipos de almacenes de claves: protegido por software y protegido por HSM. La diferencia puede ser confusa, por lo que quiero aclararlo antes de comenzar:

- Un *almacén protegido por software* almacena claves, secretos y certificados en un HSM, pero todas las operaciones criptográficas que se requieren para cifrar o descifrar su contenido las realiza la plataforma Azure en el software. Los almacenes protegidos por software son excelentes para escenarios de desarrollo y pruebas, aunque puede decidir que las cargas de trabajo de producción requieren una forma ligeramente más segura de realizar las operaciones criptográficas.
- Un *almacén protegido por HSM* almacena claves, secretos y certificados en un HSM, y las operaciones criptográficas que se requieren para cifrar o descifrar su contenido se realizan directamente en el HSM. También puede generar sus propias claves seguras en un HSM local y, a continuación, importarlas a Azure. Hay algunas herramientas y procesos adicionales que seguir, pero de esta manera se asegura de que usted tiene el control absoluto de las claves y que nunca saldrán del límite de HSM.

Para maximizar la seguridad y la integridad de sus datos, los almacenes protegidos por hardware son el enfoque preferido para las cargas de trabajo de producción.

Independientemente del tipo de almacén que utilice, es importante recordar que todos sus datos se almacenan de forma segura en un HSM con el estándar federal de procesamiento de información (FIPS) 140-2 nivel 2 validado (como mínimo), y que Microsoft no puede acceder o recuperar sus claves. Hay un costo adicional para que los almacenes protegidos por HSM, así como con cualquier cosa en Azure y la informática en la nube, equilibren el costo frente al riesgo de que sus datos se vean comprometidos.

15.1.2 Creación de un almacén de claves y secreto

Un almacén digital suena muy bien, pero puede sentirse un poco inseguro respecto a cómo hacer uso de la energía que proporciona Azure Key Vault. Creemos un ejemplo de un servidor básico que ejecuta una base de datos como el servidor MySQL, como se muestra en la figura 15.3.

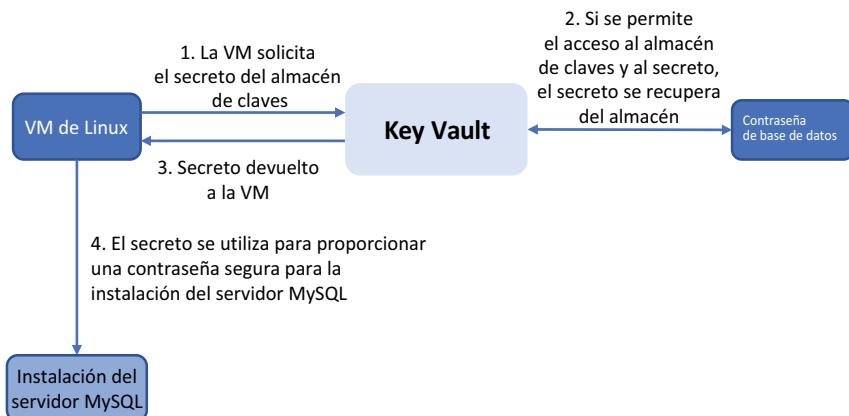


Figura 15.3 En los ejercicios siguientes, creará un ejemplo de un secreto almacenado en un almacén de claves que se puede utilizar como la contraseña de la base de datos para una instalación del servidor MySQL. Se crea una VM que tiene permisos para solicitar el secreto desde el almacén de claves. Luego, el secreto recuperado se utiliza para introducir automáticamente una credencial segura durante el proceso de instalación de la aplicación.

Uno de los primeros ejercicios de este libro fue crear una VM y luego instalar la pila del servidor web LAMP. Es probable que se le haya solicitado una contraseña para el servidor MySQL, o se utilizó automáticamente una contraseña en blanco. Ahora que sabe todo acerca de los almacenes de claves, puede recuperar de manera automática una contraseña del almacén y usarla dinámicamente para instalar y configurar el servidor.

Pruébelo ahora

Para crear un almacén de claves y agregar un secreto, complete los siguientes pasos:

- Abra Azure Portal; inicie Cloud Shell; y cree un grupo de recursos, como `azuremolchapter15`:

```
az group create --name azuremolchapter15 --location eastus
```

- Cree un almacén de claves con un nombre único, como `azuremol`, y habilítelo para la implementación a fin de que pueda utilizarlo para injectar claves y certificados en una VM:

```
az keyvault create \
--resource-group azuremolchapter15 \
```

```
--name azuremol \
--enable-soft-delete \
--enabled-for-deployment
```

De forma predeterminada, a su cuenta de usuario Azure se le asignan permisos completos para el almacén de claves. Para estos ejercicios, eso está bien, aunque como un procedimiento recomendado de seguridad debe considerar limitar quién puede acceder a su almacén de claves. Puede agregar el parámetro `--no-self-perms` para omitir la asignación de permisos a su cuenta.

- 3 Cree un secreto, como `databasenamepassword`, y asigne un valor de contraseña, como `SecureP@ssw0rd`. (Sí, realmente seguro, ¿cierto?) Este secreto se puede utilizar como las credenciales para un servidor de base de datos, que se implementará en los siguientes ejercicios:

```
az keyvault secret set \
--name databasenamepassword \
--vault-name azuremol \
--description "Database password" \
--value "SecureP@ssw0rd"
```

- 4 Tiene permisos completos para el almacén de claves, para que pueda ver el contenido de su secreto:

```
az keyvault secret show \
--name databasenamepassword \
--vault-name azuremol
```

Desde una perspectiva de administración, también puede realizar acciones comunes como hacer una copia de seguridad y restaurar, descargar, actualizar y eliminar los elementos almacenados en un almacén de claves. Una propiedad adicional que configuró cuando se creó el almacén de claves es la opción de `enable-soft-delete`. Si sus aplicaciones y servicios no pueden recuperar los secretos que necesitan del almacén de claves, es posible que deba lidiar con una enorme interrupción de la aplicación. Un almacén de claves puede almacenar metadatos para secretos hasta 90 días después de que realmente se eliminan, lo que le permite recuperar datos que se eliminan de forma incorrecta o malintencionada.

- 5 Elimine la clave que acaba de crear para simular un error, o posiblemente alguien con intenciones malintencionadas:

```
az keyvault secret delete \
--name databasenamepassword \
--vault-name azuremol
```

- 6 Recupere el secreto para que pueda continuar utilizando la contraseña de la base de datos con su aplicación y servicios:

```
az keyvault secret recover \
--name databasenamepassword \
--vault-name azuremol
```

Si realmente desea eliminar un secreto, también tiene la opción de purgar un secreto eliminado. Esta opción elimina permanentemente el secreto sin esperar a que transcurra el período de recuperación predeterminado de 90 días.

No dude en volver a usar `az keyvault secret show` para ver la información sobre su secreto y confirmar que la contraseña que guardó está allí. Ahora, continuaremos para ver cómo una VM puede acceder a un almacén de claves y utilizar el secreto para instalar el servidor MySQL.

15.2 Identidades administradas para recursos de Azure

La capacidad de utilizar Azure Key Vault para almacenar secretos o claves es excelente, pero ¿cómo acceder a estos secretos? La CLI de Azure o Azure PowerShell pueden acceder a la información almacenada en un almacén de claves, pero a menudo es más conveniente permitir que las máquinas virtuales o las aplicaciones recuperen directamente los secretos o las claves cuando las necesiten. Una manera de hacerlo es con identidades administradas para los recursos de Azure, como se muestra en la figura 15.4.

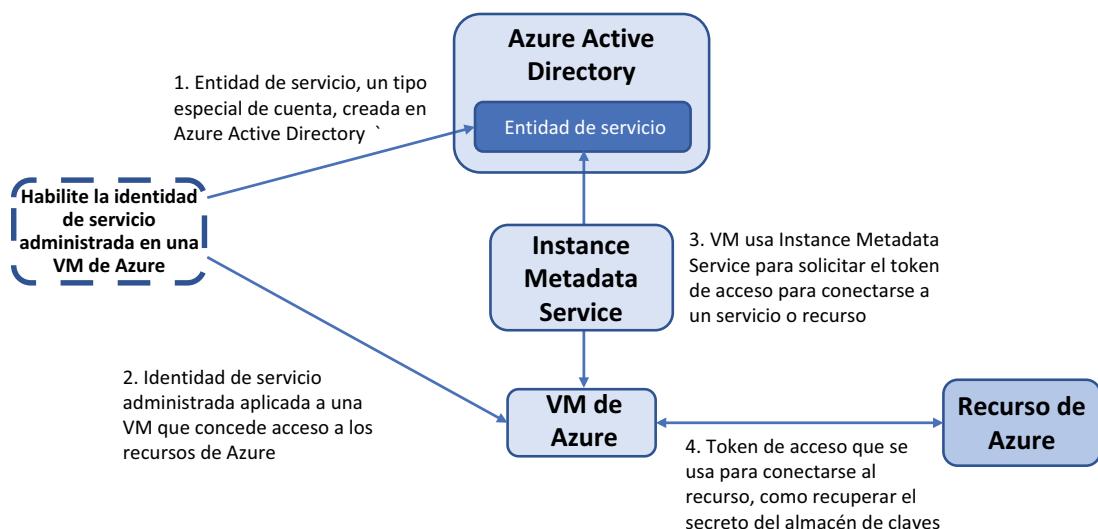


Figura 15.4 Cuando se crea una identidad administrada para una VM, se crea una entidad de servicio en Azure Active Directory. Esta entidad de servicio es un tipo especial de cuenta que se puede utilizar para que los recursos se autentiquen. Luego, esta VM utiliza el punto de conexión de Instance Metadata Service para hacer las solicitudes de acceso a los recursos. El punto de conexión se conecta a Azure AD para solicitar tokens de acceso cuando la VM necesita solicitar datos de otros servicios. Cuando se devuelve un token de acceso, se puede utilizar para solicitar acceso a los recursos de Azure, como un almacén de claves.

Una *identidad administrada* le permite crear un tipo especial de cuenta que puede usar un recurso de Azure, como una VM. Si ha utilizado un servicio de directorio como Active Directory, a menudo se utiliza una cuenta de equipo para identificar y conceder acceso a varios recursos de red que necesita un equipo. Usted no crea y utiliza cuentas de usuario habituales para este tipo de autenticación, lo que mejora la seguridad: por ejemplo, se puede conceder un conjunto restrictivo de permisos a un único equipo en lugar de preocuparse por los permisos de usuario y el acceso a carpetas compartidas.

Una identidad administrada es como una cuenta de equipo, pero se almacena en Azure Active Directory (Azure AD). La identidad, denominada *entidad de servicio*, es exclusiva de cada máquina virtual y se puede utilizar para asignar permisos a otros recursos de Azure, como una cuenta de Azure Storage o almacén de claves. La VM tiene permisos para acceder a esos recursos, por lo que puede realizar tareas de script (como con Azure Automation, que exploraremos en el capítulo 18) que no requieren intervención del usuario ni avisos para los nombres de usuario y contraseñas. Las máquinas virtuales se autentican y la plataforma Azure autoriza el acceso a sus recursos asignados.

Puede crear dos tipos de identidades administradas:

- *Asignada por el sistema*: este tipo de identidad administrada se aplica directamente a un recurso, como una máquina virtual, y solo la utiliza ese recurso. Cada recurso cuenta con su propia identidad cuando se trata de auditar o solucionar el acceso. Cuando se elimina el recurso, la identidad administrada se elimina automáticamente.
- *Asignada por el usuario*: se crea y administra un recurso Azure independiente para la identidad administrada especificada. Esta identidad administrada puede compartirse con otros recursos para definir el acceso. Cuando se elimina cualquier recurso que use la identidad, la identidad administrada sigue estando disponible para su uso.

Veamos cómo se puede utilizar una identidad administrada asignada por el sistema para solicitar el secreto databasepassword desde un almacén de claves. Una vez que la VM puede recuperar el secreto, la contraseña se puede utilizar para instalar automáticamente un servidor de base de datos MySQL. Con un almacén de claves y MSI, puede ejecutar un par de comandos para recuperar el secreto del almacén de claves, ejecutar el instalador del servidor MySQL y proporcionar automáticamente la contraseña segura.

Azure Instance Metadata Service

Una VM habilitada con una identidad administrada utiliza un punto de conexión REST a través de Instance Metadata Service (IMDS) para solicitar un token de acceso de Azure AD que luego puede utilizar para solicitar datos de Azure Key Vault. Pero, ¿qué es Instance Metadata Service?

IMDS es un punto de conexión REST que solo es accesible internamente para las máquinas virtuales. El punto de conexión está disponible en la dirección no enrutable de

169.254.169.254. Una VM puede hacer una solicitud al punto de conexión IMDS para recuperar información sobre sí misma, como la región Azure o el nombre del grupo de recursos. Esta capacidad permite a la VM entender cómo y dónde se está ejecutando la plataforma Azure. Se puede acceder al punto de conexión IMDS desde muchos lenguajes, incluidos Python, C#, Go, Java y PowerShell.

Para los eventos de mantenimiento, también se puede consultar el punto de conexión IMDS para que la VM tenga conocimiento de una actualización pendiente o de un evento de reinicio. Luego, pueden llevarse a cabo las tareas previas a la actualización o el reinicio. Dado que IMDS es un punto de conexión REST en una dirección IP no enrutable, no hay ningún agente o extensión para que instale la VM, y no hay problemas de seguridad de red o enrutamiento.

A efectos de la identidad administrada, el punto final de IMDS se utiliza para transmitir la solicitud de un token de acceso a Azure AD. Este enfoque proporciona una manera segura para que las máquinas virtuales soliciten acceso sin necesidad de hablar directamente con Azure AD.

Pruébelo ahora

Para crear una VM con una MSI, complete los pasos siguientes:

- 1 Cree una VM de Ubuntu; a continuación, proporcione su grupo de recursos, como `azuremolchapter15` y dé un nombre a la VM, como `molvm`. Se crea una cuenta de usuario denominada `azuremol`, y las claves SSH que ha utilizado en capítulos anteriores se agregan a la VM:

```
az vm create \
--resource-group azuremolchapter15 \
--name molvm \
--image ubuntults \
--admin-username azuremol \
--generate-ssh-keys
```

- 2 Como un procedimiento recomendado de seguridad, no debe permitir que las cuentas accedan a todos los recursos a través de toda su suscripción a Azure. Especialmente para las identidades administradas, solo conceda la cantidad mínima de permisos necesarios.

Para este ejercicio, abarque el acceso solo a su grupo de recursos, como `azuremolchapter15`. Se establece el alcance mediante la consulta del identificador del grupo de recursos con `--query id`. A continuación, este identificador se asigna a una variable llamada `scope`:

```
scope=$(az group show --resource-group azuremolchapter15
      --query id --output tsv)
```

- 3 Cree una identidad administrada asignada por el sistema para la VM con el rol de lector para que solo pueda leer recursos, no hacer cambios en ellos. Abarque la identidad en el grupo de recursos. Se proporciona la variable que creó en el paso anterior que contiene el identificador de grupo de recursos:

```
az vm identity assign \
--resource-group azuremolchapter15 \
--name molvm \
--role reader \
--scope $scope
```

- 4 Aplique los permisos en Azure Key Vault que concede el acceso al principal de servicio para la identidad administrada. Puede hacerlo mediante el portal bajo Directivas de acceso para el recurso de Key Vault, o bien puede utilizar la CLI de Azure. Utilicemos la CLI para ver cómo obtener la información mediante programación.

Primero, obtenga información sobre la entidad de servicio de Azure AD para su identidad administrada. Filtre por `display-name` de la VM que creó en el paso 3, como `molvm`:

```
az ad sp list \
--display-name molvm \
--query [].servicePrincipalNames
```

La salida es similar al siguiente ejemplo condensado. No se preocupe demasiado por lo que significan estos valores; no tiene que trabajar con ellos más allá de asignar los permisos iniciales aquí. Una vez más, puede utilizar Azure Portal para evitar la CLI si no se siente cómodo.

Tome nota del primer `servicePrincipalName`. Este valor se utiliza para asignar permisos en los recursos de Azure, como su almacén de claves y se requiere en el siguiente paso:

```
[  
  "887e9665-3c7d-4142-b9a3-c3b3346cd2e2",  
  "https://identity.azure.net//  
  ↗ihxXtwZEiAeNXU8eED2Ki6FXRPkk1thh84S60CiqA4="  
]
```

- 5 Ahora defina la directiva de acceso en el almacén de claves de modo que la entidad de servicio de su VM pueda leer secretos e ingrese su primer `servicePrincipalName` del paso 4:

```
az keyvault set-policy \
--name azuremol \
--secret-permissions get \
--spn 887e9665-3c7d-4142-b9a3-c3b3346cd2e2
```

Un punto que señalar aquí es que cuando se creó la identidad administrada y se abarcó al grupo de recursos, eso no significó que la VM podría entonces hacer lo que quisiera. En primer lugar, la única función creada para la identidad era leer los permisos de los recursos. Pero todavía tenía que asignar permisos al propio almacén de claves. Estas capas de seguridad y permisos le dan un control preciso sobre los recursos exactos a los que puede acceder cada identidad.

Ahora que tiene acceso a un almacén de claves, es probable que desee saber cómo recuperar el secreto, ¿cierto?

15.3 Obtención de un secreto dentro de una máquina virtual con identidad de servicio administrado

Ha almacenado un secreto en un almacén de claves para una contraseña de base de datos, y tiene una VM con una identidad administrada que proporciona acceso para leer ese secreto en el almacén de claves. ¿Y ahora qué? ¿Cómo recupera el secreto y lo usa? En la figura 15.5 se muestra cómo una VM utiliza IMDS para solicitar acceso a un recurso, como un almacén de claves. Recorramos los pasos para ver cómo la VM recupera el secreto.

La mayoría de los casos prácticos de Azure Key Vault no tendrían una VM conectándose y recuperando los secretos de esta manera. Key Vault realmente se destaca cuando las propias aplicaciones, dentro del código, llegan a recuperar los secretos. El código de la aplicación utilizaría el SDK de Azure apropiado, como Python, .Net o Java. Para evitar la complejidad del código que abstrae lo que está sucediendo, el siguiente ejercicio utiliza una VM y algo de trabajo de línea de comandos. Mientras trabaje en este ejercicio, recuerde que esta magia suele ocurrir dentro del código de la aplicación.

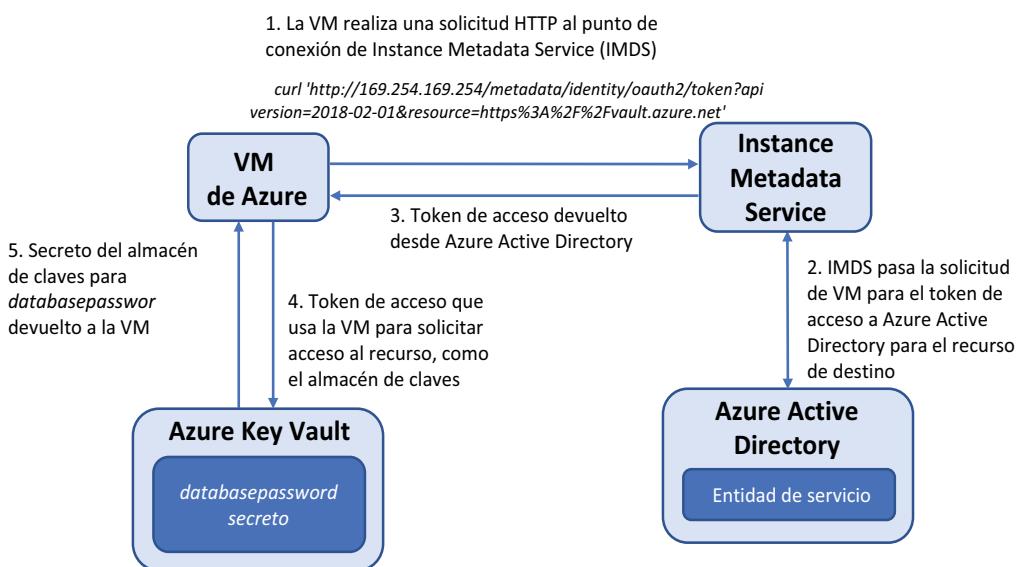


Figura 15.5 La VM utiliza IMDS para solicitar acceso a un almacén de claves. El punto de conexión se comunica con Azure AD para solicitar un token de acceso. El token de acceso se devuelve a la VM, que luego se utiliza para solicitar acceso desde el almacén de claves. Si el almacén de claves concede el acceso, el secreto para `databasenamepassword` se devuelve a la VM.

Pruébelo ahora

Para recuperar y utilizar un secreto en una VM con una identidad administrada, complete los pasos siguientes:

- 1 Obtenga la dirección IP pública de la VM que creó en el ejercicio anterior, como molvm:

```
az vm show \
--resource-group azuremolchapter15 \
--name molvm \
--show-details \
--query [publicIps] \
--output tsv
```

- 2 SSH para su VM, como ssh azuremol@publicIps.
- 3 Para acceder a un almacén de claves, necesita un token de acceso. Este token de acceso se solicita desde IMDS. Es una solicitud HTTP y en una VM de Linux puede utilizar el programa curl para hacer la solicitud. IMDS pasa su solicitud a AAD:

```
curl 'http://169.254.169.254/metadata/identity/oauth2/token?
api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net'
-H Metadata:true
```

- 4 La salida es un poco difícil de leer, porque se ve como un texto desordenado. Está en el formato de JSO Web Token (JWT). Para procesar la salida JSON y permitir que las cosas sean más legibles para el ser humano, instale un analizador JSON llamado jq:

```
sudo apt-get update && sudo apt-get -y install jq
```

- 5 Haga su solicitud curl otra vez, pero ahora vea la salida con jq:

```
curl 'http://169.254.169.254/metadata/identity/oauth2/token?
api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net'
-H Metadata:true --silent | jq
```

En estos primeros pasos se muestra cómo se hacen las solicitudes y cuál es la apariencia de la salida, como se muestra en la figura 15.6. Si todavía inicia sesión en la VM y solicita manualmente un token de acceso, ¿qué sentido tiene utilizar una identidad administrada? Podría simplemente proporcionar sus propias credenciales. En el uso de producción, es probable que utilice un script que se ejecuta en la VM para hacer la solicitud de un token de acceso automáticamente y, a continuación, recuperar el secreto del almacén de claves. Seguiremos adelante para ver cómo automatizar este proceso y recuperar el secreto.

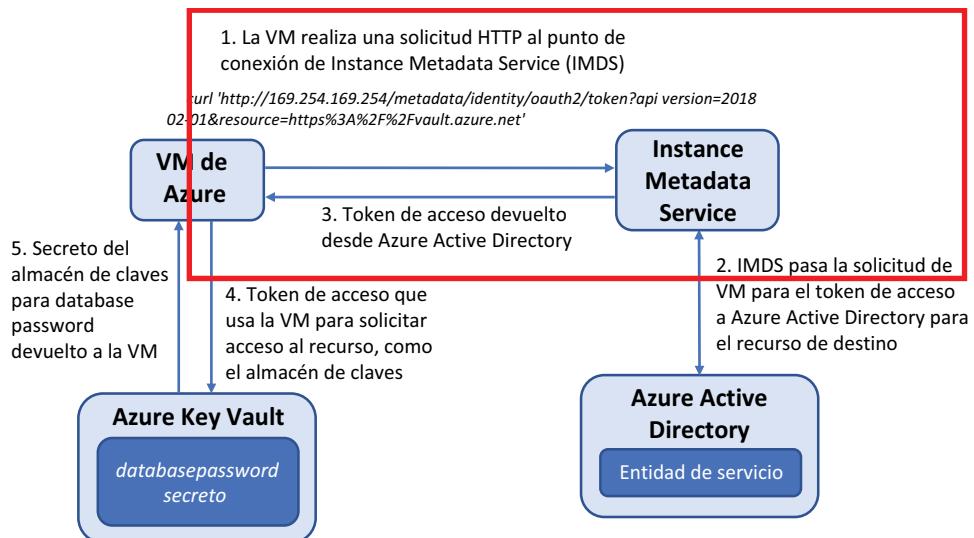


Figura 15.6 La solicitud curl cubre los tres primeros pasos de este diagrama. Se realiza la solicitud curl, el punto de conexión se comunica con Azure AD y se emite un token de acceso.

- 6 Para facilitar las cosas, y si iba a hacer todo esto en un script, puede utilizar jq para procesar la respuesta curl, extraer solo el token de acceso y establecerlo como una variable denominada access_token:

```
access_token=$(curl
  ↪ 'http://169.254.169.254/metadata/identity/oauth2/token?
  ↪ api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net'
  ↪ -H Metadata:true --silent | jq -r '.access_token')
```

- 7 Como paso manual para ayudarlo a entender cómo se ve esto, vea la variable access_token:

```
echo $access_token
```

- 8 ¡Ahora la parte divertida! Utilice el token de acceso para solicitar su secreto desde el almacén de claves. Primero, hagamos esto manualmente para que entienda lo que sucede.

- 9 Recupere el secreto con otra solicitud curl, y dé formato a la salida con jq. Introduzca su propio nombre de almacén de claves al principio de `https:// address`:

```
curl https://azuremol.vault.azure.net/secrets/databasepassword?
  ↪ api-version=2016-10-01 -H "Authorization: Bearer $access_token"
  ↪ --silent | jq
```

La salida es similar a lo siguiente, que muestra el valor de la contraseña almacenada en el secreto, junto con algunos metadatos adicionales sobre el secreto sobre los que no necesita preocuparse por ahora:

```
{
  "value": "SecureP@ssw0rd!",
  "contentType": "Database password",
  "id":
  ➔ "https://azuremol.vault.azure.net/secrets/databasepassword/
  ➔ 87e79e35f57b41fdb882c367b5c1ffb3",
}
```

Esta solicitud curl es la segunda parte del flujo de trabajo, como se muestra en la figura 15.7.

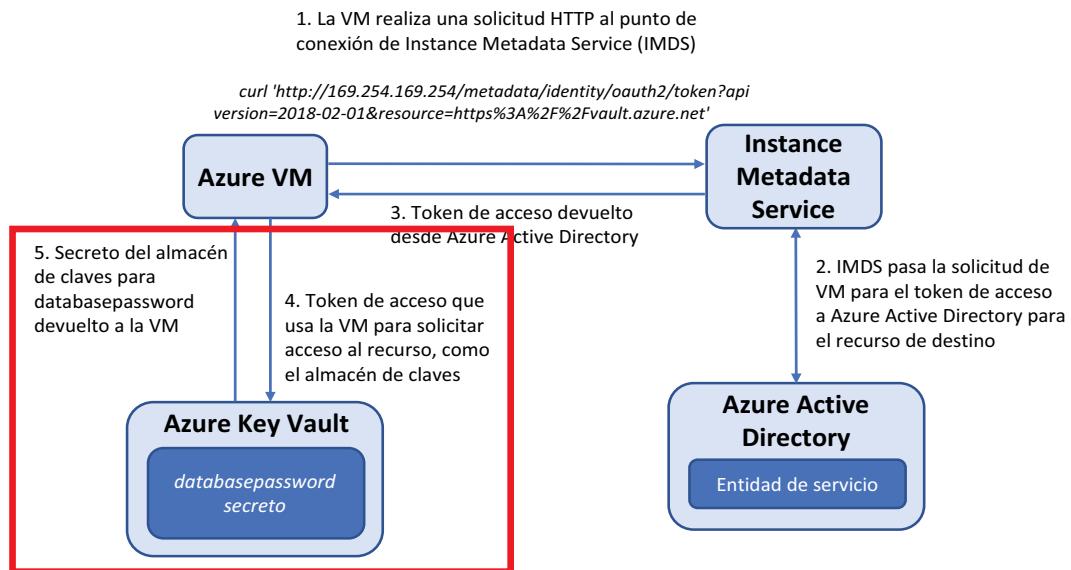


Figura 15.7 Esta segunda solicitud curl cubre los dos últimos pasos del diagrama. El token de acceso se utiliza para solicitar el secreto desde el almacén de claves. Se devuelve la respuesta JSON, que incluye el valor del secreto.

- 10 De la misma manera que usó una variable para almacenar el token de acceso, en un script también puede asignar el valor del secreto a una variable. Esta vez, utilice jq para procesar la respuesta, extraer solo el secreto del valor y establecerlo como una variable denominada database_password:

```
database_password=$(curl
➔ https://azuremol.vault.azure.net/secrets/databasepassword?
➔ api-version=2016-10-01 -H "Authorization: Bearer $access_token"
➔ --silent | jq -r '.value')
```

- 11 Una vez más, como paso manual para ayudarle a entender el proceso, vea el contenido de la variable database_password:

```
echo $database_password
```

Espero que nos siga. Si, por ejemplo, escribe una aplicación en Python, ASP.NET o Node.js, el proceso será similar al realizar una solicitud para el token de acceso y, a continuación, utilizar el token para solicitar un secreto desde un almacén de claves. Hay otras bibliotecas que podría utilizar en su código en lugar de la utilidad jq de la línea de comandos.

Como resumen rápido, todos estos pasos pueden condensarse en dos líneas, como se muestra en la siguiente lista.

Listado 15.1 Solicitud de un token de acceso y luego un secreto desde un almacén de claves

```
access_token=$(curl  
  ↪ 'http://169.254.169.254/metadata/identity/oauth2/token?  
  ↪ api-version=2018-02-01&resource=https%3A%2F%2Fvault.azure.net'  
  ↪ -H Metadata:true --silent | jq -r '.access_token')  
database_password=$(curl  
  ↪ 'https://azurem01.vault.azure.net/secrets/databasepassword?  
  ↪ api-version=2016-10-01 -H "Authorization: Bearer $access_token"  
  ↪ -silent | jq -r '.value')
```

¿Y ahora qué? La identidad administrada para su VM puede recuperar un secreto de un almacén de claves. Veamos cómo puede utilizar esa identidad administrada para instalar y configurar MySQL Server.

En Ubuntu, puede configurar selecciones de configuración para los instaladores de paquetes, como el servidor MySQL. Estas selecciones de configuración le permiten proporcionar valores como nombres de usuario y contraseñas, y usarlas automáticamente en la parte pertinente del proceso de instalación. Se terminan las solicitudes manuales para proporcionar una contraseña, como puede haber visto en el capítulo 2.

- 12 Defina las selecciones de configuración para las contraseñas del servidor MySQL con la variable database_password que creó en el paso 10:

```
sudo debconf-set-selections <<< "mysql-server mysql-server/root_password  
  ↪ password $database_password"  
sudo debconf-set-selections <<< "mysql-server mysql-  
  ↪ server/root_password_again password $database_password"
```

- 13 Instale el servidor MySQL. No hay avisos, porque las selecciones de configuración proporcionan la contraseña:

```
sudo apt-get -y install mysql-server
```

- 14 Demostremos que todo esto funcionó. Visualice la variable database_password para que pueda ver claramente cuál debe ser su contraseña:

```
echo $database_password
```

- 15 Inicie sesión en el servidor MySQL. Cuando se le solicite una contraseña, introduzca el valor de database_password, que es el valor del secreto del almacén de claves:

```
mysql -u root -p
```

Inició sesión en el servidor MySQL, lo que confirma que el secreto del almacén de claves se usó para crear con éxito las credenciales de SQL Server.

- 16 Escriba exit dos veces para cerrar el símbolo del sistema del servidor MySQL, y luego cierre su sesión SSH en la VM.

Este ejemplo es básico, y todavía necesitaría proteger el servidor MySQL y proporcionar credenciales adicionales para que las aplicaciones accedan a bases de datos o tablas. La ventaja de utilizar un secreto desde un almacén de claves es que usted garantiza que todas las contraseñas son las mismas. Por ejemplo, si utiliza conjuntos de escala de máquinas

virtuales, cada instancia de VM puede solicitar automáticamente el secreto e instalar el servidor MySQL para que esté listo para servir los datos de la aplicación. Esas contraseñas nunca se definen en scripts y nadie necesita ver cuáles son las contraseñas. Incluso podría generar contraseñas al azar y rotarlas como secretos en un almacén de claves.

Almacenar contraseñas en un almacén de claves está muy bien, pero ¿puede utilizar un almacén de claves para almacenar certificados y recuperarlos automáticamente desde sus aplicaciones o máquinas virtuales? ¡Claro que puede!

15.4 Creación e inyección de certificados

Los certificados digitales son una forma común de seguridad y autenticación en los servicios y aplicaciones web. Una entidad de certificación (CA) emite los certificados, la que (esperamos) sea de confianza para los usuarios finales. El certificado les permite a los usuarios comprobar que un sitio web o aplicación es realmente lo que dice que es. Cada vez que ve un sitio web con una dirección de navegador web que comienza con https:// and has a padlock symbol, the traffic is encrypted and se protege mediante un certificado digital.

La administración de certificados digitales puede convertirse en una tarea de administración importante. Un problema común es cómo almacenar y conceder acceso a certificados, ya que los servicios y las aplicaciones los necesitan. En los ejercicios anteriores, examinamos cómo se puede utilizar un almacén de claves para compartir secretos y claves seguros con servicios y aplicaciones, pero un almacén de claves también puede hacer lo mismo con certificados. Como se muestra en la figura 15.8, se puede utilizar un almacén de claves para solicitar, emitir y almacenar certificados.

En el uso de producción, siempre debe utilizar una CA de confianza para emitir sus certificados. Para uso interno, puede emitir certificados autofirmados que usted mismo cree.

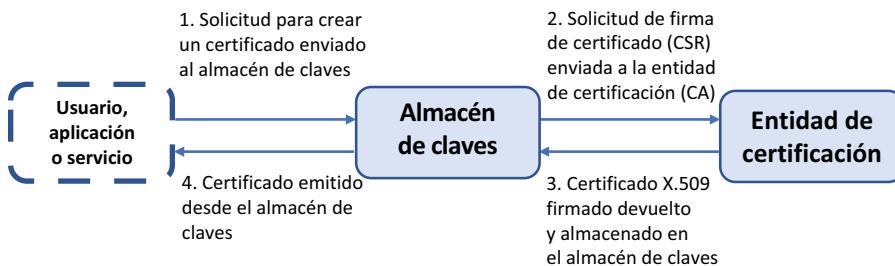


Figura 15.8 Un usuario, una aplicación o un servicio pueden solicitar un nuevo certificado desde un almacén de claves. El almacén de claves envía una solicitud de firma de certificado (CSR) a una CA externa de terceros o a una CA interna de confianza. Azure Key Vault también puede actuar como su propia CA para generar certificados autofirmados. A continuación, la CA emite un certificado X.509 firmado, que se almacena en el almacén de claves. Por último, el almacén de claves devuelve el certificado al solicitante original.

Estos certificados autofirmados no son de confianza para otros servicios y aplicaciones, por lo que normalmente generan una advertencia, pero los certificados autofirmados le permiten ponerse en marcha rápidamente y asegurarse de que el código funciona según lo esperado con el tráfico cifrado.

Azure Key Vault puede generar certificados autofirmados para usted. Por debajo, Key Vault actúa como su propia CA para solicitar, emitir y almacenar certificados. Usemos esta capacidad para generar un certificado autofirmado y ver cómo inyectarlo fácilmente en una VM. El certificado se utiliza para que un servidor web básico le muestre cómo habilitar SSL rápidamente para proteger su tráfico web.

Pruébelo ahora

Para crear e inyectar un certificado en una VM, complete los pasos siguientes:

- 1 Cree un certificado autofirmado en Azure Key Vault y escriba un nombre, como molcert. Las directivas se utilizan para definir propiedades como períodos de caducidad, intensidad de cifrado y formato de certificado. Puede crear diferentes directivas para adaptarse a las necesidades de sus aplicaciones y servicios. Para este ejercicio, utilice la directiva predeterminada que crea un certificado de 2048 bits y es válido durante un año:

```
az keyvault certificate create \
    --vault-name azuremol \
    --name molcert \
    --policy "$(az keyvault certificate get-default-policy)"
```

- 2 Para ver el certificado en acción, cree otra VM, como molwinvm. Esta vez, cree una VM de Windows que utiliza Windows Server 2019, a fin de que extienda el encanto del sistema operativo y vea que estas funciones de Key Vault no dependen de un sistema operativo específico. Proporcione su propio nombre de usuario y contraseña de administrador:

```
az vm create \
    --resource-group azuremolchapter15 \
    --name molwinvm \
    --image win2019datacenter \
    --admin-username azuremol \
    --admin-password P@ssw0rd1234
```

- 3 Puede agregar automáticamente el certificado a la VM directamente desde la CLI de Azure. Este enfoque no está basado en una identidad administrada; la plataforma Azure inyecta el certificado mediante el agente de VM de Windows Azure.

Agregue su certificado, como molcert, a la VM que creó en el paso 2, como molwinvm:

```
az vm secret add \
    --resource-group azuremolchapter15 \
    --name molwinvm \
    --keyvault azuremol \
    --certificate molcert
```

- 4 Conéctese a la VM y compruebe que el certificado se inyectó correctamente. Para conectarse a su VM, primero obtenga su dirección IP pública:

```
az vm show \
    --resource-group azuremolchapter15 \
    --name molwinvm \
    --show-details \
```

```
--query [publicIps] \
--output tsv
```

Utilice un cliente de conexión de Escritorio remoto de Microsoft en el equipo para conectarse a la VM. Utilice las credenciales para conectarse a localhost\azuremol, no las credenciales predeterminadas del equipo local que su cliente de Escritorio remoto puede intentar utilizar, como se muestra en la figura 15.9.

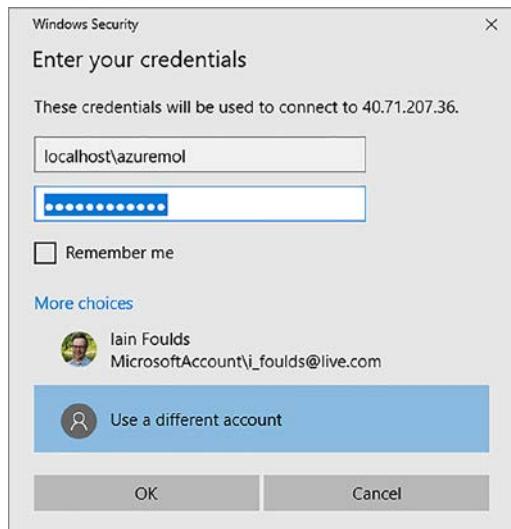


Figura 15.9 El cliente de Escritorio remoto puede intentar utilizar las credenciales de equipo local predeterminadas. En su lugar, seleccione Usar otra cuenta y, a continuación, proporcione las credenciales localhost\azuremol que especificó al crear la VM.

- 5 Cuando haya iniciado sesión, seleccione el botón Inicio de Windows, escriba mmc y abra la Microsoft Management Console.
- 6 Seleccione Archivo > Agregar o quitar complemento y, a continuación, seleccione la opción para agregar el complemento Certificados.
- 7 Seleccione para agregar certificados para la cuenta de equipo, seleccione Siguiente y, a continuación, Finalizar.
- 8 Seleccione Aceptar para cerrar la ventana Agregar o quitar complemento.
- 9 Expanda la carpeta Certificados (equipo local) > Personal > Certificados. Se enumera el certificado de Azure Key Vault que inyectó en la VM; por ejemplo, CLIGetDefaultPolicy, como se muestra en la figura 15.10.

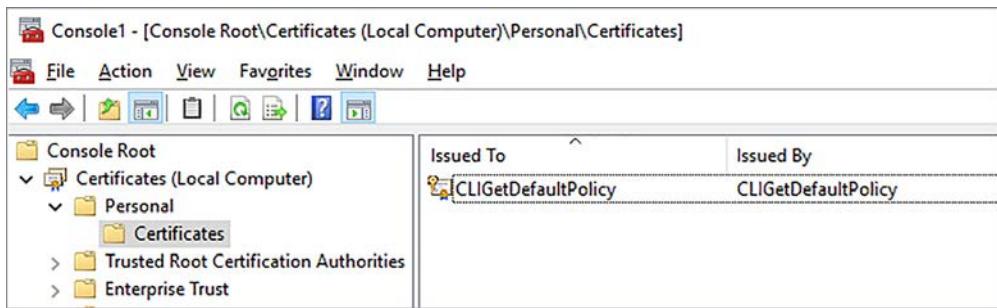


Figura 15.10 En la Microsoft Management Console, agregue el complemento Certificados en el equipo local. Expanda el almacén Personal > Certificados para ver los certificados instalados. Se muestra el certificado inyectado desde Key Vault.

¡Eso es todo! Cree el certificado en Key Vault y, a continuación, agréguelo a la VM. El certificado se coloca en el almacén de certificados local del equipo, lo que permite que cualquier servicio o aplicación acceda a este. En una VM de Windows, los certificados se almacenan en la memoria caché de certificados local, como se ve en este ejercicio. En las máquinas virtuales de Linux, los archivos .prv y .crt para las partes privadas y públicas del certificado se almacenan en /var/lib/waagent/. Puede mover los certificados a donde los necesite para su aplicación o servicio.

Los certificados pueden utilizarse para la autenticación entre clientes y servidores, o entre componentes de aplicaciones y servicios. Un ejemplo común es que un servidor web utilice un certificado SSL, que es lo que hará en el laboratorio del fin del capítulo.

15.5 Laboratorio: Configuración de un servidor web seguro

En el último ejercicio, se inyectó un certificado autofirmado de Azure Key Vault en una VM de Windows. Para este laboratorio, instale y configure el servidor web IIS para utilizar el certificado, siguiendo esta guía:

- 1 Abra PowerShell en la VM de Windows e instale el servidor web de IIS:

```
Add-WindowsFeature Web-Server -IncludeManagementTools
```
- 2 Abra el Administrador de Internet Information Server (IIS). Puede hacerlo desde el menú Herramientas del Administrador de servidores.
- 3 Para el sitio web predeterminado, elija Modificar enlaces.
- 4 Agregue un vínculo HTTPS en todas las direcciones IP no asignadas en el puerto 443.
- 5 Seleccione el certificado autofirmado que creó e inyectó desde Key Vault, normalmente con el nombre CLIGetDefaultPolicy.
- 6 Abra un navegador web en la VM y escriba `https://localhost`. Generó un certificado autofirmado en Key Vault, por lo que el navegador web no confía en él.
- 7 Acepte la advertencia para continuar y compruebe que el enlace HTTPS funciona.
- 8 De nuevo en Azure Cloud Shell o el portal, cree una regla de NSG para la VM en el puerto TCP 443. Escriba `https://yourpublicipaddress` en el navegador web de su equipo local. Esta es la experiencia que sus usuarios recibirán, con una advertencia sobre un certificado autofirmado que no es de confianza. Para la mayoría de los casos de uso, recuerde utilizar una entidad de CA interna o de terceros para generar certificados de confianza y almacenarlos en un almacén de claves.

Azure Security Center y actualizaciones

¿No sería genial que Azure fuera lo suficientemente inteligente como para monitorear todos sus recursos de aplicaciones básicos y alertarlo en caso de que surgieran inquietudes de seguridad? ¿O qué ocurriría si su negocio tuviera directivas de seguridad ya definidas? (Si no tiene directivas de seguridad, ¡deje de leer de inmediato y comience a crear algunas!). En este último caso, ¿cómo puede asegurarse de que sus implementaciones de Azure siguen cumpliendo la normativa? Si alguna vez se ha sometido a una auditoría de seguridad de TI, sabe lo divertido que puede ser examinar una lista de configuraciones erróneas aplicadas a su entorno, en especial los lapsos de seguridad básicos que sabe que debe evitar.

Azure Security Center ofrece una ubicación central que agrupa las alertas de seguridad y las recomendaciones para su revisión. Puede definir sus propias directivas de seguridad y luego dejar que Azure monitoree el estado de sus recursos para velar por el cumplimiento.

En este capítulo, analizaremos cómo Security Center puede alertarlo en caso de problemas y proporcionar los pasos para corregirlos, cómo puede usar el acceso a VM Just-In-Time para controlar y realizar auditorías de las conexiones remotas, y cómo Update Management mantiene actualizadas automáticamente sus VM con los parches de seguridad más recientes.

16.1 Azure Security Center

En el transcurso de este libro hemos analizado temas relacionados con la seguridad, tales como cómo crear y configurar grupos de seguridad de red (NSG) para restringir el acceso a las VM y cómo permitir únicamente el tráfico cifrado a las cuentas de Azure Storage. Para sus propias implementaciones más allá de los ejercicios en este libro, ¿cómo sabe dónde comenzar y cómo puede comprobar que ha aplicado todos los procedimientos recomendados de seguridad? Allí es donde Azure Security Center puede ayudarlo, al revisar las áreas de su entorno que pudiera haber olvidado.

Azure Security Center escanea sus recursos, recomienda correcciones y ayuda a solucionar las inquietudes de seguridad, tal como se muestra en la figura 16.1. Cuando solo tiene algunas VM de prueba y una sola red virtual en su suscripción de Azure, no parece tan difícil llevar un registro de las restricciones de seguridad que debe implementar. Sin embargo, a medida que escala a decenas, cientos o miles de VM, llevar un registro manual de las configuraciones de seguridad que deben aplicarse a cada VM se vuelve inmanejable.

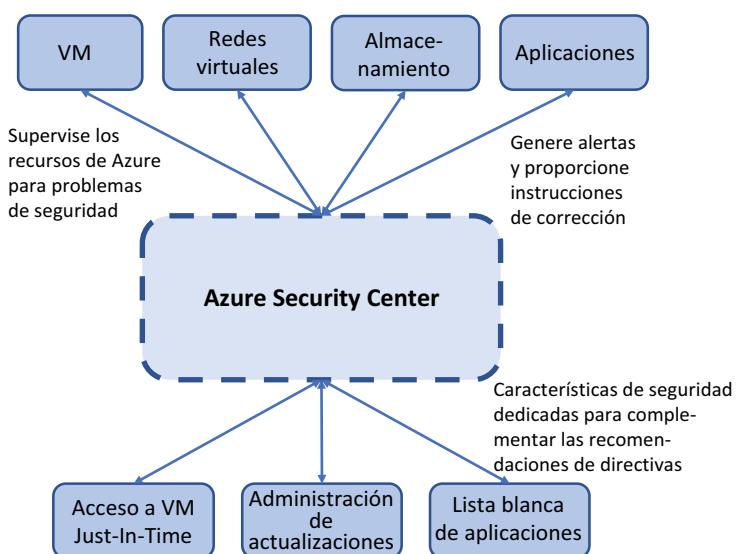


Figura 16.1 Azure Security Center monitorea sus recursos de Azure y usa directivas de seguridad definidas para alertarlo ante posibles amenazas y vulnerabilidades. Se proporcionan recomendaciones y pasos para solucionar problemas. También puede usar el acceso a VM Just-In-Time, monitorear y aplicar actualizaciones de seguridad, y controlar las aplicaciones que se encuentran en la lista de permitidos y pueden ejecutarse en las VM.

Security Center también puede alertarlo sobre los procedimientos recomendados generales, tales como si una VM no tiene diagnósticos habilitados. ¿Recuerda cuando en el capítulo 12 analizamos cómo monitorear y solucionar problemas de las VM? Debe instalar y configurar el agente de diagnóstico *antes* de que tenga un problema. Si sospecha que se produjo una infracción de seguridad, es posible que no pueda tener acceso a la VM y los registros de revisión. Sin embargo, si hubiera configurado la extensión de diagnóstico para transmitir los registros a Azure Storage, habría podido revisar lo que ocurría y, si todo va bien, haber localizado el origen y la magnitud del problema.

Pruébelo ahora

Para comenzar con Azure Security Center, complete los pasos siguientes:

- 1 Abra el Azure Portal y seleccione el icono de Cloud Shell en el menú superior.
- 2 Cree un grupo de recursos; proporcione un nombre, como `azuremolchapter16`; y proporcione una ubicación, como `eastus`:

```
az group create --name azuremolchapter16 --location eastus
```

- 3 Cree una VM básica de Linux para que Security Center tenga algo para monitorear y para lo que pueda proporcionar recomendaciones:

```
az vm create \
--resource-group azuremolchapter16 \
--name azuremol \
--image ubuntults \
--admin-username azuremol \
--generate-ssh-keys
```

- 4 Cuando se haya implementado la VM, cierre Cloud Shell.
- 5 En Azure Portal, seleccione Security Center en la lista de servicios que se encuentra a la izquierda. La primera vez que se abra el panel, este tardará unos segundos en preparar todos los componentes disponibles; consulte la figura 16.2.

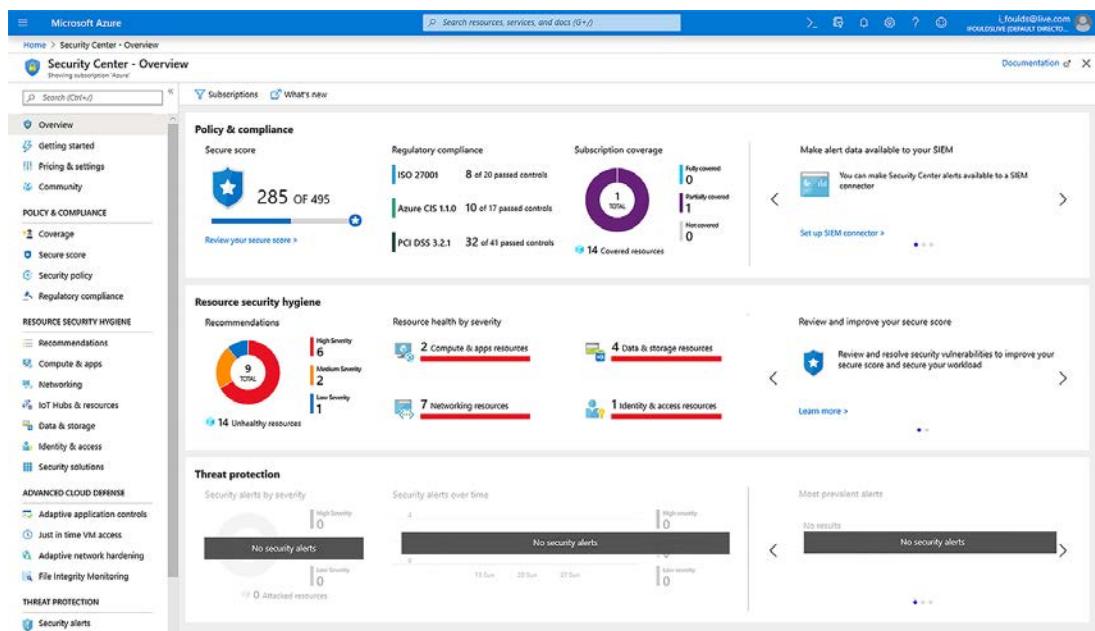


Figura 16.2 La ventana Información general de Azure Security Center ofrece una lista de recomendaciones, alertas y eventos. Puede seleccionar un tipo de recurso básico como Informática o Redes para ver una lista de los elementos de seguridad específicos de esos recursos.

Security Center analiza cómo se implementan recursos como VM, reglas de NSG y almacenamiento. Los puntos de referencia de seguridad incorporados se usan para identificar problemas y proporcionar recomendaciones. Por ejemplo, la red virtual implementada con su VM genera un par de advertencias, como se muestra en la figura 16.3. Puede, y debiera, implementar sus propias directivas de seguridad que le indiquen a Azure de qué forma desea restringir el acceso o lo que se debe hacer para cumplir con las exigencias del negocio. Luego, a medida que crea o actualiza recursos, Azure

The screenshot shows the Azure Security Center interface for a specific subnet. At the top, it displays 'Home > Security Center - Networking > azuremolVNET > azuremolSubnet'. Below this, there's a summary section for 'Resource health' showing 'azuremolSubnet' with a total of 1 recommendation. The recommendations are categorized by priority: High (1), Medium (0), and Low (0). A red bar indicates the severity of the high-priority recommendation. Below this, the 'information' section provides details about the subnet, including its name ('Resource Name: azuremolSubnet'), group ('Resource Group: azurermolchapter16'), and subscription ('Subscription: Azure'). The final section, 'Recommendation list', contains one entry: 'Subnets should be associated with a Network Security Group', marked as 'High' priority.

Figura 16.3 La red virtual de su máquina virtual ya activa las advertencias de seguridad. En este ejemplo, advierte que se debe asociar un grupo de seguridad de red a la subred.

monitorea continuamente las desviaciones de estas directivas y lo alerta sobre las medidas que deben tomarse para corregir los problemas de seguridad. Utilizará las directivas de seguridad predeterminadas de Azure en este capítulo, sin embargo, piense en cualquier configuración de seguridad específica que pudiera desear aplicar a sus VM y cómo estas podrían definirse en sus propias directivas personalizadas.

- 6 Elija Proceso y aplicaciones en el menú de la izquierda de la ventana Security Center; a continuación, elija VM y equipos.
- 7 Seleccione la VM que creó en el paso 3. Aunque acaba de crear esta VM y usó los valores predeterminados de la CLI de Azure, se muestran algunas advertencias de seguridad.

Explore algunas de estas recomendaciones. Al seleccionar cada recomendación, algunas solo le dan más información; otras lo guían en la corrección. No se trata de reglas rígidas, sino de recomendaciones y procedimientos recomendados. En su propio entorno, algunas de ellos pueden no tener sentido. Pero son un buen punto de partida para saber qué cosas debería hacer para asegurar los recursos mientras los crea en Azure.

16.2 Acceso Just-In-Time

En la sección 16.1, aprendió cómo Security Center sugiere limitar el alcance de la conectividad remota entrante. Podría proporcionar un intervalo de IP para limitar el tráfico, pero idealmente, usted solo abre la conectividad entrante cuando es necesario. De esa manera, la VM se cierra completamente para las conexiones remotas y solo se puede acceder a ella por un periodo breve cuando es

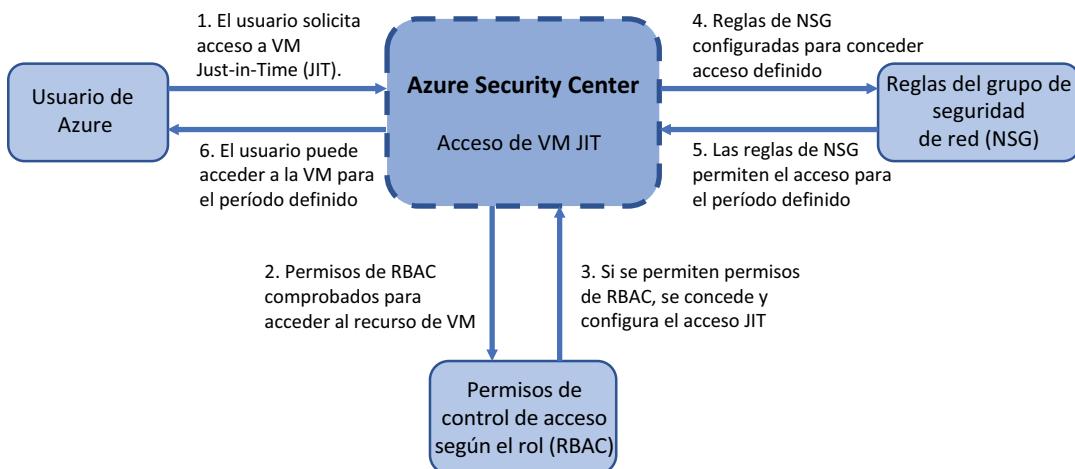


Figura 16.4 Con el acceso a VM Just-In-Time, las reglas de NSG se configuran para denegar las conexiones remotas a una VM. Los permisos de RBAC se usan para verificar los permisos cuando un usuario solicita acceso a una VM. Estas solicitudes se auditán y, si se concede la solicitud, las reglas de NSG se actualizan para permitir el tráfico desde un intervalo de IP determinado, por un período definido. El usuario puede acceder a la VM solo durante este período. Cuando haya caducado el período, las reglas de NSG automáticamente se revierten al estado denegar.

necesario. Y sí, aún debe limitar esa breve ventana de conectividad a un intervalo de IP específico. Ahí es donde el acceso a VM Just-In-Time resulta útil, tal como se muestra en la figura 16.4.

Con el acceso JIT, Security Center ajusta de forma dinámica las restricciones de acceso a una VM. Cuando están activadas, se crean reglas de NSG que deniegan todo el tráfico de conexión remota. Entonces, un usuario solo puede solicitar acceso a una VM cuando es necesario. En combinación con el control de acceso basado en roles (analizado en el capítulo 6), Security Center determina si un usuario tiene derechos de acceso a una VM cuando solicita una conexión. Si el usuario tiene permisos, Security Center actualiza las reglas de NSG pertinentes para permitir el tráfico entrante. Estas reglas solo se aplican por una ventana de tiempo específica. Cuando el período se acaba, las reglas se revierten y la VM nuevamente se cierra a las conexiones remotas. Si tiene una conexión activa a una VM, no se desconecta automáticamente cuando expira el tiempo. Puede terminar su trabajo de mantenimiento o solución de problemas y desconectarse cuando esté listo, pero no podrá iniciar una nueva conexión a menos que solicite de nuevo el acceso JIT.

Beber desde una boca de incendios

No hemos examinado realmente Azure Firewall, pero es un recurso de red virtual que es un poco más parecido a un firewall físico local que a los NSG por sí mismos. Si necesita más flexibilidad y control de tráfico, Azure Firewall es una gran opción, aunque lleva asociado un costo.

Sin profundizar demasiado en Azure Firewall, quiero señalar que Azure Security Center también puede integrarse con Azure Firewall para abrir y cerrar las reglas necesarias. Si

utiliza Azure Firewall para proteger el tráfico de las VM en las redes virtuales, no solo en las NSG, aún puede usar la administración automatizada de las normas de acceso a las máquinas virtuales JIT.

Para obtener más información sobre Azure firewall, consulte la documentación en <https://docs.microsoft.com/azure/firewall/overview>.

¿Cuándo usaría JIT en su pizzería de ficción? Piense en cualquier VM que ejecutaría su aplicación web, su sistema de pedidos o sus aplicaciones lógicas de negocio. ¿Desea que estos estén conectados a Internet y estén disponibles para que las personas tengan acceso a ellos en todo momento? Espero que no. Existen motivos válidos para el acceso remoto con SSH o RDP, sin embargo, siempre trate de minimizar la cantidad de tiempo que ese acceso está disponible. Aunque tenga reglas de NSG que restrinjan el acceso a ciertos intervalos de IP, JIT agrega otro nivel de protección en términos de a qué pueden tener acceso los usuarios de Azure, y luego crea un rastro de auditoría más fácil sobre el cual Security Center puede proporcionar informes.

Pruébelo ahora

Para permitir el acceso a VM Just-In-Time, complete los pasos siguientes:

- 1 Abra Azure Portal y seleccione Security Center en el menú que se encuentra a la izquierda.
- 2 En Defensa avanzada de la nube, seleccione Acceso a VM Just-In-Time.
- 3 Si se le solicita, elija la opción Probar el acceso a VM Just-In-Time o Actualizar al nivel estándar de Security Center. Esta prueba gratuita dura 60 días y no se debe extender automáticamente. Se superpone con su cuenta gratuita de Azure y su uso no tiene costo. Seleccione la opción Aplicar plan estándar y luego espere unos momentos para habilitarlo. Cuando esté activado, es posible que deba cerrar y volver a abrir Azure Portal antes de poder completar los pasos siguientes.
- 4 Vuelva a seleccionar Acceso a VM Just-In-Time en la ventana Security Center. Cuando esté activada la cuenta de nivel estándar, puede ver una lista de VM para usar.
- 5 Seleccione su VM y, a continuación, elija Solicitar acceso, como se muestra en la figura 16.5.

Virtual machine ↑↓	Approved	Last access ↑↓	Connection details
<input checked="" type="checkbox"/>  azuremol	0 Requests	N/A	

Figura 16.5 Seleccione una VM de las opciones que se encuentran en Recomendado y luego elija Activar JIT en 1 VM. Actualmente, el estado muestra que esta VM está Abierta para todo el acceso remoto, que indica la gravedad de la inquietud de seguridad como Alta.

De forma predeterminada, JIT define las reglas que pueden abrir los puertos para SSH (puerto 22), RDP (puerto 3389) y comunicación remota de PowerShell (puertos 5985 y 5986) por un periodo de tres horas.

- 6 Para este ejercicio, elija habilitar SSH desde su propia IP. Como procedimiento recomendado para el uso en producción, introduzca una justificación para tener constancia de por qué se solicita el acceso. Deje todas las configuraciones predeterminadas y elija Abrir puertos, tal como se muestra en la figura 16.6.

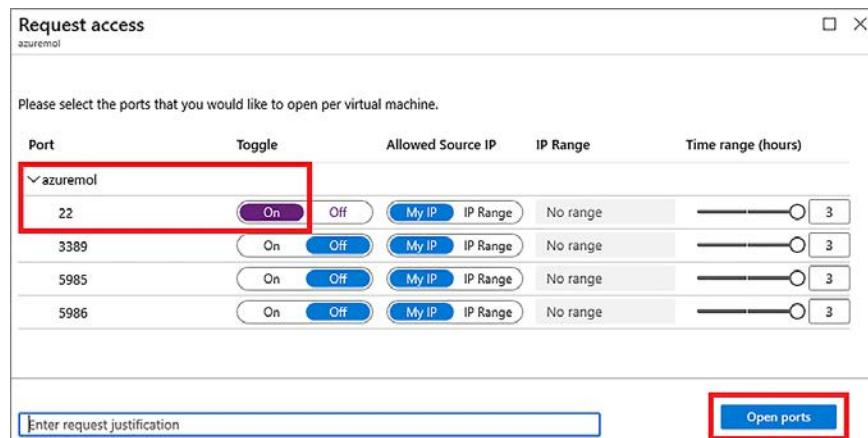


Figura 16.6 Cuando activa JIT, puede cambiar las reglas predeterminadas que se deben permitir, las IP de origen permitidas y un tiempo máximo de solicitud en horas. Estas reglas de JIT facilitan un control granular de lo que se permite, para permitir solo el mínimo de conectividad.

- 7 Con JIT habilitado, desplácese hasta su grupo de recursos y seleccione su VM.
- 8 Elija Redes para ver la configuración de red virtual asignada para la VM. Se muestra la lista de reglas de NSG asignadas, tal como en la figura 16.7.

Priority	Name	Port	Protocol	Source	Destination	Action	
100	SecurityCenter-JITRule-1115349600-87...	22	Any	73.254.183.78	10.0.0.4	<input checked="" type="radio"/> Allow	...
1000	default-allow-ssh	22	TCP	Any	Any	<input checked="" type="radio"/> Allow	...
65000	AllowVnetInbound	Any	Any	VirtualNetwork	VirtualNetwork	<input checked="" type="radio"/> Allow	...
65001	AllowAzureLoadBalancerInbound	Any	Any	AzureLoadBalancer	Any	<input checked="" type="radio"/> Allow	...
65500	DenyAllInbound	Any	Any	Any	Any	<input checked="" type="radio"/> Deny	...

Figura 16.7 Las reglas de JIT se crean con la prioridad más baja. Estas prioridades aseguran que las reglas de JIT tengan prioridad sobre cualquier regla posterior que se aplique en el nivel de subred.

Las reglas de JIT se muestran en la parte superior de la lista, ya que tienen la prioridad más baja. Se permite el tráfico a la dirección IP de la VM, pero solo desde su propia dirección IP. Esto es lo que configuró JIT. Lo que puede parecer extraño aquí es que aún existe una regla default-allow-ssh que permite todo el tráfico. Piense en el capítulo 5, cuando hablamos de las NGS. ¿Puede decir qué está pasando aquí?

JIT solo se aplica a la VM. En la regla de JIT, Destino muestra la dirección IP de la VM. En el ejemplo que se muestra en la figura 16.7, esta es 10.0.0.4. Se permite el tráfico. Sin embargo, la regla de NSG real se aplica a toda la subred. La regla default-allow-ssh se aplica en el nivel de subred y permite el tráfico desde cualquier origen y hacia cualquier destino.

Las reglas de NSG se procesan el orden de prioridad, de baja a alta. Como se analizó en el capítulo 5, una acción Denegación siempre tiene efecto, independientemente de cualquier regla adicional. Aunque se cambiara esa regla default-allow-ssh para denegar el tráfico, la regla JIT seguiría permitiendo el acceso a la VM específica y desde la dirección IP de origen definida.

Tenga cuidado con esta superposición de normas de NSG. Lo ideal sería eliminar la regla default-allow-ssh y, a continuación, permitir el acceso solo según se requiera con JIT. En este enfoque, la regla final DenyAllInbound deniega SSH. Cuando necesite conectarse a una VM, utilice JIT para solicitar el acceso, que crea automáticamente una regla para permitir SSH con alcance a su dirección IP durante un período definido.

La regla de NSG se elimina automáticamente una vez transcurrido el período especificado. De forma predeterminada, las reglas de JIT se aplican durante tres horas. Después de ese período, la VM regresa a un estado más seguro y usted deberá volver a solicitar acceso a la VM.

Este proceso de JIT controla quién puede solicitar, y a quién se le puede conceder, acceso a la VM. Sin embargo, el solo hecho de que una persona pueda solicitar acceso con éxito a una VM no significa que tenga permisos para iniciar sesión en esa VM. Todo lo que ocurre en Azure es que se actualizan las reglas de NSG definidas. Security Center y JIT no pueden agregar, eliminar ni actualizar las credenciales de acceso en la VM.

También se registran todas las solicitudes de JIT. En Security Center, seleccione la opción Acceso de VM Just in Time y luego elija su regla. A la derecha, seleccione la opción de menú ... y luego elija Registro de actividad. Este registro de actividad le ayuda a auditar quién solicitó acceso a una VM en caso de un problema.

El acceso a VM JIT es una forma en la que Security Center y Azure ayudan a mantener sus VM seguras. Controlar el acceso a las VM es gran parte de la seguridad. Pero, ¿qué pasa con las aplicaciones, bibliotecas y servicios que se ejecutan en las máquinas virtuales? Ahí es donde debe asegurarse de que se apliquen las actualizaciones de seguridad más recientes a sus VM de forma oportuna.

16.3 Azure Update Management

Un área que puede informar Azure Security Center es el estado de cualquier actualización del SO que requiera la VM. En la pizzería, debiera intentar instalar los parches de seguridad y aplicaciones más recientes. Usted no desea ejecutar sistemas que tienen una vulnerabilidad o área de ataque conocida, por lo que una forma de automatizar las actualizaciones de esos sistemas y rastrear el seguimiento mejora la seguridad. Cuando trabaja con aplicaciones que implican los datos y la información de

pago del cliente, no ejecuta sistemas que no tengan instalados los parches más recientes. Además, recuerde planificar un entorno de prueba que le permita aplicar los parches de seguridad de forma segura y validar que estos no provoquen problemas antes de aplicarlos a los sistemas de producción.

Las VM de Azure cuentan con Update Management, una característica de administración de actualizaciones que puede escanear, informar y reparar las actualizaciones del SO. Lo bueno de esta solución es que funciona en Windows y Linux, e incluso dentro de Linux, entre diferentes distribuidores como Ubuntu, Red Hat y SUSE. En la figura 16.8 se muestra cómo Update Management monitorea y puede instalar las actualizaciones necesarias.

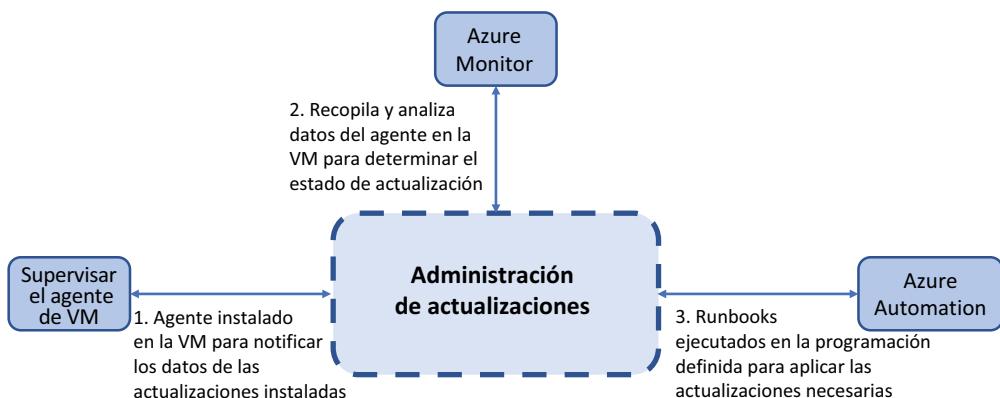


Figura 16.8 Update Management instala un agente de VM que recopila información sobre las actualizaciones instaladas en cada VM. Estos datos son analizados por Azure Monitor y se vuelven a informar a la plataforma Azure. Entonces, se puede programar la instalación automática de la lista de actualizaciones necesarias a través de runbooks de Azure Automation.

La VM tarda unos minutos en prepararse y volver a informar su estado de actualización, así que vamos a configurar la VM y luego veremos lo que ocurre en segundo plano.

Pruébelo ahora

Para configurar la VM para Update Management, complete los pasos siguientes:

- 1 Abra Azure Portal y elija Grupos de recursos en el menú que se encuentra a la izquierda.
- 2 Seleccione el grupo de recursos, como `azuremolchapter16`, y luego seleccione su VM, como `azuremol`.
- 3 En Operaciones, seleccione Update Management.
- 4 Acepte la opción predeterminada en Ubicación y la opción para crear una cuenta de espacio de trabajo de Log Analytics y Automation. Examinaremos en detalle estos componentes en lo que resta de esta sección.
- 5 Para activar la administración de actualización de la VM, seleccione Activar.

Regresará a la ventana Información general de Update Management, sin embargo, esta tardará unos minutos en configurar la VM y volver a informar su estado. Siga leyendo y deje que el proceso continúe.

Echemos un vistazo más a lo que se hace para que esta solución de Update Management funcione.

16.3.1 Servicios combinados de administración de Azure

Si ha trabajado con tecnologías de Microsoft locales, es posible que se haya encontrado con el conjunto de aplicaciones de System Center. System Center consta de varios componentes, tales como Configuration Manager, Operations Manager, Orchestrator y Data Protection Manager. Tiene un par de partes más, sin embargo, esos componentes principales proporcionan una forma de hacer lo siguiente:

- Definir las configuraciones y el estado deseado
- Instalar aplicaciones y actualizaciones
- Informar sobre el estado y la seguridad
- Automatizar las implementaciones de servicios y aplicaciones de gran tamaño
- Hacer copias de seguridad y replicar datos

Durante los últimos años, a medida que los negocios han ido migrando a la informática en la nube, los servicios de Azure que pueden funcionar en un entorno híbrido reemplazaron aquellos componentes locales más tradicionales de System Center. En capítulos anteriores analizamos dos componentes, a pesar de que no se haya dado cuenta:

- *Azure Backup* ofrece una forma de hacer copias de seguridad de las VM o de archivos individuales, definir directivas de retención y restaurar datos.
- *Azure Site Recovery* le permite replicar VM en diferentes regiones geográficas, en caso de un desastre natural o una interrupción prolongada.

Tanto Azure Backup como Site Recovery le ayudaron a proteger sus datos en el capítulo 13. Ahora usará algunos servicios adicionales con Update Management:

- Los espacios de trabajo de *Log Analytics* recopilan información de diversas fuentes o agentes, y le permiten definir directivas y consultas para alertarlo sobre las condiciones que podrían producirse. Estas consultas y alertas pueden ayudarlo a rastrear el estado de actualización de una VM o notificarlo de problemas de configuración o seguridad.
- *Azure Monitor* detalla e informa acerca de la información basada en el procesamiento que se lleva a cabo en los espacios de trabajo de Log Analytics. Azure Monitor ofrece una forma centralizada de ver alertas, consultar datos de registro y generar notificaciones en todos sus recursos de Azure.
- *Azure Automation* le permite desarrollar runbooks que ejecutan comandos o scripts enteros. Los runbooks pueden ser implementaciones grandes y complejas, que pueden llamar a varios otros runbooks. Analizaremos en detalle Azure Automation en el capítulo 18.

En la figura 16.9 se muestra la integración de estos componentes.

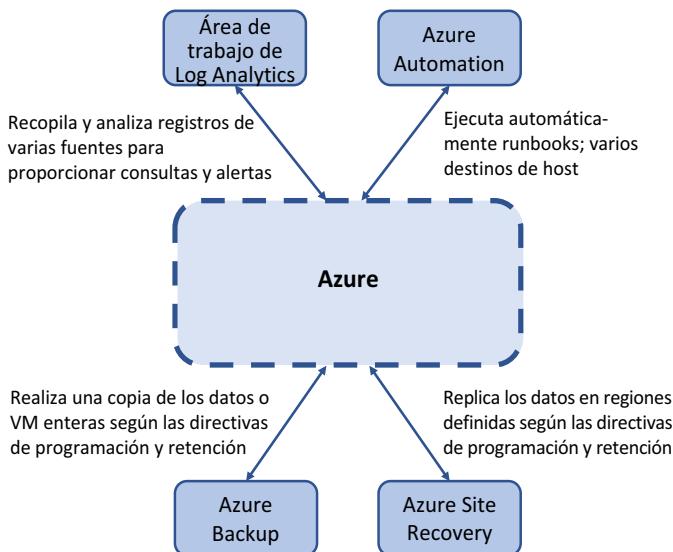


Figura 16.9 Varios servicios de Azure trabajan en conjunto para proporcionar características de administración y configuración en todo el entorno de la aplicación. Los servicios que usan estos componentes no se limitan a las VM o los recursos de Azure y pueden trabajar con otros sistemas de proveedores en la nube o locales cuando se configuran correctamente.

Tanto los espacios de trabajo de Log Analytics como Azure Automation son componentes potentes que fácilmente podrían tener sus propios libros. Con solo algunas VM que administrar, puede encontrar sencillo pasar por alto la necesidad de un repositorio de registro centralizado para consultas y alertas, o una forma de automatizar las configuraciones e implementaciones en las VM. Si aún no ha hecho una lista de los componentes de Azure a los que debe realizar seguimiento cuando termine este libro, comience una y agregue ambos componentes a esa lista.

Algo que debe entender es que, en Azure, con frecuencia hay múltiples servicios y componentes que pueden interactuar y complementarse entre sí. Del mismo modo que las VM de Azure y las redes virtuales de Azure son servicios individuales, ambos servicios también se complementan entre sí o incluso se basan el uno en el otro. Azure Backup y la extensión de diagnósticos de Azure son componentes individuales excelentes, sin embargo estos realmente se destacan si se usan los espacios de trabajo de Log Analytics y Azure Monitor para monitorear su estado y recopilar cualquier evento o advertencia que se genere. Espero que haya comenzado a identificar algunos de estos componentes relacionados y ve cómo los servicios de Azure a menudo se complementan entre sí. Ahora que estamos en los últimos capítulos analizando las opciones de seguridad y supervisión, el objetivo es asegurar que las aplicaciones que ejecuta en Azure están en buen estado y son estables.

Esta cosa llamada “identidad”

Al pensar en los servicios que se complementan entre sí, una gran (y quiero decir *¡gran!*) parte de Azure de la que solo hemos hablado superficialmente es Azure Active Directory (Azure AD). La identidad es el centro de todo en Azure, y Azure AD ofrece algunas de las características de seguridad que analizamos en el capítulo 6 con el modelo de implementación de Azure Resource Manager. La posibilidad de usar RBAC para limitar las acciones que ciertos usuarios o grupos pueden realizar con un recurso está vinculada a una solución de identidad central. Incluso el hecho de poder iniciar sesión en Azure Portal o la CLI de Azure es impulsado por Azure AD.

En este libro no se aborda Azure AD porque el alcance de lo que ofrece es amplio y difiere bastante de los servicios de IaaS y PaaS de Azure como VM, conjuntos de escala y aplicaciones web. Es posible que se produzcan coincidencias en el público de los temas, sin embargo, la mayoría de los desarrolladores tendría un objetivo de aprendizaje diferente con respecto a Azure AD en comparación con los de un administrador de aplicaciones o un profesional de TI que implementa la infraestructura.

En función de su cuenta de Azure, también podría verse limitado con respecto a lo que puede hacer con Azure AD. Cuando se registra para obtener una cuenta de prueba gratuita de Azure, se crea para usted una instancia de AAD predeterminada. Usted es la cuenta principal en ese directorio, por lo que tiene derechos de administrador completos. Si inicia sesión en Azure con una cuenta de su negocio o institución educativa, hay una buena posibilidad de que tenga pocos derechos administrativos o que no los tenga. Así que aunque pudieramos coincidir en algunos temas, es posible que no pueda realizar los ejercicios directamente. Y realmente no le recomiendo indagar en un entorno de Azure AD real para aprender cómo funcionan las cosas.

Sin embargo, Azure AD es otro de esos servicios centrales de Azure que vincula a muchos otros servicios y componentes. La informática en la nube no hace que las cosas sean más fáciles por arte de magia ni acaba con silos operacionales, aún necesita las habilidades para trabajar con diferentes equipos y partes interesadas. Espero que en el transcurso de estos capítulos haya captado las habilidades básicas para estos servicios de Azure, los que le ayudarán a entender cómo desarrollar aplicaciones grandes y redundantes, además de transmitir en un mejor nivel y con más conocimientos lo que otros equipos pueden enfrentar.

16.3.2 Revisión y aplicación de actualizaciones

Puede que el agente de VM tarde un tiempo en realizar el primer análisis e informe sobre el estado de las actualizaciones aplicadas. La lista de componentes instalados también debe compararse con la lista de actualizaciones disponibles para un SO y una versión determinados. Si su VM no ha terminado e informado sobre su estado, siga leyendo y vuelva a revisar en unos minutos. Cuando esté listo, la descripción general se verá como la figura 16.10. Sea paciente; puede tardar de 10 a 15 minutos para que la preparación del agente aparezca como Preparada y le permita programar actualizaciones para la instalación.

Es genial contar con una lista de las actualizaciones requeridas, ¿pero qué hay de la forma de instalarlas? ¡Ahí es cuando Azure Automation hace su ingreso! Cuando habilitó Update Management, se crearon varios runbooks de Azure Automation que automáticamente manipularon el proceso para aplicar las actualizaciones requeridas.

Update name	Classification	Information link
apport	Others	
file	Others	
libbind2-0	Others	
libmagic-mgc	Others	
libmagic1	Others	
libwbclient0	Others	
libxslt1.1	Others	
python-samba	Others	
python3-apport	Others	
python3-problem-report	Others	
samba-common	Others	
samba-common-bin	Others	
samba-libs	Others	

Figura 16.10 Cuando el agente de VM ha realizado el análisis de cumplimiento, se proporciona una lista de las actualizaciones disponibles. En función del SO y la versión, Update Management puede trabajar con el espacio de trabajo de Log Analytics y Azure Monitor para clasificar las actualizaciones basándose en la gravedad, o para proporcionar vínculos a las páginas de revisiones de actualizaciones pertinentes.

Pruébelo ahora

Si tiene suerte (o no lo tiene), su VM podría informar que no se requieren actualizaciones. Las imágenes de VM se actualizan frecuentemente en Azure, y si usted implementa una VM poco después de la creación de la imagen más reciente, todas las actualizaciones necesarias ya están instaladas. Si es así, lea detenidamente estos pasos para que entienda qué se necesita cuando su VM debe actualizarse.

Para aplicar las actualizaciones necesarias para su VM, complete los pasos siguientes:

- 1 En la sección Update Management de su VM, seleccione Programar la implementación de actualización.
- 2 Dé un nombre a la implementación de actualización, como `azuremolupdates`, y luego revise las clasificaciones de actualización. Puede controlar qué conjuntos de actualizaciones se aplican. Por ahora, deje todas las opciones predeterminadas.
- 3 Actualizaciones para excluir le permite especificar las actualizaciones que no desea instalar. Si sabe que su aplicación requiere una versión específica de un paquete o una biblioteca, puede asegurarse de que no se haya instalado un paquete actualizado que interrumpa las cosas. Revise las opciones disponibles, pero no hay nada que cambiar en este ejercicio

- 4 Seleccione Programar configuración y luego elija el momento de aplicación de las actualizaciones en las opciones de calendario y hora. La hora de inicio debe ser al menos cinco minutos después de la hora actual, para así dar a la plataforma Azure unos momentos para procesar y programar el runbook en Azure Automation.
- 5 Cuando esté listo, seleccione Aceptar.
- 6 Si algunas aplicaciones y servicios deben detenerse o cerrarse antes de aplicar las actualizaciones y volver a iniciarse cuando estas finalicen, elija Scripts previos + Scripts posteriores. Se pueden configurar tareas de automatización independientes para llevar a cabo acciones en las máquinas virtuales antes y después de aplicar las actualizaciones.
- 7 La ventana Mantenimiento (minutos) define por cuánto tiempo se puede ejecutar el proceso de actualización antes de que la VM deba volver a funcionar. Esta ventana evita los procesos de actualización prolongados que pudieran provocar que una VM no estuviera disponible por horas a la vez. Le recomendamos acortar o alargar la ventana de mantenimiento según los acuerdos de nivel de servicio que tenga para las aplicaciones que se ejecutan en esas VM o el número y el tamaño de las actualizaciones necesarias. Acepte el valor predeterminado y, a continuación, seleccione Crear.
- 8 En la ventana Update Management, seleccione Programas de implementación. Las actualizaciones se muestran como programadas para instalar en la fecha y hora que seleccionó, tal como se muestra en la figura 10.86.11.

Name	Next run time	Operating system	Scope	Recurrence	Maintenance window	...
azurem01updates	10/31/2019, 7:58 AM	Linux	azurem01	One time	120 minutes	...

Figura 16.11 Se muestra la lista de tareas de implementación programadas. Si lo desea, puede eliminar una tarea determinada; de lo contrario, las actualizaciones se aplican automáticamente a la hora definida.

- 9 En la parte superior de la ventana Update Management, seleccione Administrar varias máquinas. La ventana cambia a la cuenta de Azure Automation que se creó cuando Update Management se habilitó para la VM. Por ahora, no se preocupe demasiado por lo que hacen los runbooks. No hay nada que deba personalizar, examinaremos Azure Automation en el capítulo 18.

Tenga en cuenta que puede elegir Agregar VM de Azure o Agregar una máquina que no es de Azure, tal como se muestra en la figura 16.12. Esta capacidad destaca un enfoque único de administración de actualizaciones en todo su entorno de aplicaciones, no solo para VM de Azure.

The screenshot shows the 'Update management' section of the Azure Automation portal. It displays various metrics: Non-compliant machines (0 out of 3), Machines need attention (3) with categories Critical and security (0), Other (1), and Not assessed (2); Missing updates (13) with categories Critical (0), Security (0), and Others (13); Failed update deployments (0 out of 0 in the past six months). There are buttons for 'Add Azure VMs', 'Add non-Azure machine', and 'Manage machines'. Below these are tabs for 'Machines (3)', 'Missing updates (13)', 'Deployment schedules', and 'History'. A search bar and filters for 'Machine name', 'Compliance', 'Platform', 'Operating system', and 'Critical missing up...' are also present.

Figura 16.12 En la cuenta de Azure Automation, puede administrar varios equipos y ver el estado o aplicar las actualizaciones. Tanto las VM de Azure como los equipos que no son de Azure pueden ser monitoreados y controlados con la misma cuenta de Azure Automation. En segundo plano, Azure puede integrarse con otros proveedores para instalar agentes en equipos en un entorno híbrido. Esta integración permite que un panel único y una plataforma de administración manipulen sus necesidades de actualización.

- 10 Vuelva a la ventana Update Management de su máquina virtual y seleccione la pestaña Historial. Cuando se inicia la implementación de actualizaciones, aparece su estado. Recuerde que programó el trabajo para que se ejecute unos minutos en el futuro, por lo que no aparece de inmediato.
- 11 Seleccione la programación para ver el estado y la salida, como se muestra en la figura 16.13.

The screenshot shows the 'View entire deployment run' page for a deployment named 'azuremol'. The status is 'Succeeded'. The 'Start time' was 10/31/2019, 7:59:09 AM and the 'End time' was 10/31/2019, 8:00:10 AM. The 'Update results' section shows a donut chart with 13 Updates (green), 0 Failed (red), 0 Not attempted (orange), and 0 Not selected (grey). The 'Updates status' table lists five tasks: 'apport', 'libidn2-0', 'file', 'libmagic+mgc', and 'libmagic1', all with a status of 'Succeeded'. The 'Diagnostics and Logs' section includes tabs for 'All Logs' (with a file icon), 'Output' (with a clipboard icon), and 'Errors' (with a red X icon showing 0 errors).

Figura 16.13 Puede monitorear el estado de ejecución de trabajos de Azure Automation en el portal. Para revisar o solucionar problemas con las tareas, puede hacer clic en un trabajo para ver cualquier resultado y registro generado.

- 12** Cuando finalice la implementación de la actualización, regrese a su grupo de recursos, seleccione su VM y elija Update Management. Es posible que el agente tarde unos minutos en actualizarse e informar a través del espacio de un trabajo de Log Analytics la aplicación de las actualizaciones; entonces, debiera mostrarse en el panel que la VM está actualizada y que no se requieren actualizaciones adicionales.

Este capítulo ha sido un arrollador recorrido por Security Center y componentes asociados como el acceso a VM JIT y Update Management. El objetivo es que comience a pensar más allá de cómo implementar y ejecutar una VM o aplicación web, para en cambio planificar una administración de aplicaciones más amplia que vaya con ello. La informática en la nube no cambia la necesidad de directivas de seguridad, sin duda hay una mayor necesidad de protección de recursos. Deje que las características de Azure como Security Center lo guíen con respecto a lo que se debe hacer, y use herramientas integradas como Update Management y Azure Automation para mantener la seguridad en todo momento.

16.4 **Laboratorio: habilitación de JIT y actualizaciones para una VM de Windows**

En este capítulo se trataron algunos componentes que podrían tardar un poco en habilitarse e informar sobre su estado esperado. Este laboratorio es opcional, se diseñó para demostrar que estas características no son específicas de un SO. Si no tiene tiempo o considera que entiende cómo aplicar estas características a una VM de Windows, puede omitir la realización de este laboratorio. De lo contrario, intente completar las siguientes tareas para obtener algo de práctica adicional con Security Center y Update Management. La práctica hace al maestro, ¿cierto?

- 1** Cree una VM de Windows Server de su elección en el mismo grupo de recursos que usó para los ejercicios anteriores, tales como `azuremolchapter16`.
- 2** Vea las reglas de NSG para la VM/subred, y elimine cualquier regla predeterminada que permita el RDP en el puerto TCP 3389.
- 3** Utilice el cliente local de Conexión a escritorio remoto para comprobar que las conexiones RDP están bloqueadas.
- 4** Solicite acceso JIT, vuelva a revisar las reglas de NSG y confirme que ahora puede conectar RDP a su VM.
- 5** Habilite Update Management en su VM de Windows. En esta ocasión, debería poder usar el espacio de trabajo de Log Analytics existente y las cuentas de Azure Automation.
- 6** Deje que el agente de supervisión informe sobre las actualizaciones necesarias, y luego programe las actualizaciones que se aplicarán en Azure Automation.

Parte 4

Aspectos interesantes



Ahora, lo realmente interesante! En estos últimos capítulos, aprenderá acerca de algunas de las próximas tecnologías que puede usar en Azure, tales como inteligencia artificial, machine learning, contenedores, Kubernetes y la Internet de las Cosas. Puede que no esté usando estos servicios en este momento, pero con las tendencias actuales en informática, probablemente los usará pronto. Estos servicios son algunas de las tecnologías más interesantes con las que se puede trabajar. Aunque abarquemos muy rápidamente estos temas durante su almuerzo, esta parte es una excelente forma de concluir este libro y mostrarle las posibilidades de lo que puede desarrollar en Azure.

Inteligencia artificial y machine learning

Esperamos no tener que terminar en mundos donde películas como *Terminator* y *Matrix* se vuelven realidad. En esas películas, el auge de la inteligencia artificial (IA) casi provoca la caída de la humanidad, ya que las máquinas luchan para apoderarse de sus entornos. Una causa de preocupación en la informática actual es cómo el desarrollo de la IA es llevado a cabo principalmente por grandes empresas privadas, con poca regulación (o sin ella) y una supervisión central. ¡Esto no es para nada decir que la IA es algo malo! Los asistentes digitales de los smartphones pueden ayudar en muchas tareas cotidianas. En machine learning (ML) en las aplicaciones de navegación puede supervisar la conducción diaria del usuario para sugerirle rutas alternativas en función de las condiciones de la carretera o del tiempo. Los controles de la calefacción del hogar pueden ajustarse automáticamente según la temperatura exterior, la hora del día y la época del año (como el verano o el invierno).

En el inicio de esta parte final del libro, aprenderá acerca de los servicios de Azure para machine learning e inteligencia artificial. En un capítulo. Durante el almuerzo. Vamos a establecer algunas expectativas realistas: no se convertirá en experto en ML o IA en los próximos 45 minutos. Si come rápidamente su sándwich, puede aprender lo suficiente acerca de los muchos servicios que Azure ofrece para entender cómo integrar algunos de estos servicios de ML e IA en sus aplicaciones. Para muchos de los servicios de ML e IA de Azure se espera que tenga al menos algo de experiencia previa en algoritmos de datos, lenguajes de programación, procesamiento de lotes o comprensión de lenguaje, así que no espere convertirse en un experto en la próxima hora.

En este capítulo, realizaremos un arrollador recorrido de algunos de los servicios cognitivos de Azure que ofrecen características de ML e IA. Aprenderá a usar estos servicios para ejecutar machine learning básica en modelos de datos, y luego usará un poco del servicio de Azure Web Apps y Microsoft Bot Framework para aplicar algunos de los servicios de IA que puede ejecutar el bot de una pizzería para que los clientes puedan pedir pizza.

17.1 Descripción general y relación de IA y ML

¡Sujétese fuerte, porque estamos a punto de ir de 0 a 1000 km/h en solo unas páginas! A menudo, la IA y ML se solapan a medida que desarrolla aplicaciones en Azure. Vamos a explorar en qué consiste cada una y luego nos preocuparemos de estudiar cómo trabajan juntas.

17.1.1 Inteligencia artificial

La IA permite que los equipos completen tareas con algún grado de flexibilidad y conciencia, y ajusta sus decisiones basándose en factores externos o sin la necesidad de interacción humana. El objetivo normalmente no es desarrollar un sistema completamente autónomo que pueda evolucionar y desarrollar pensamientos por sí solo, sino usar un conjunto de modelos de datos y algoritmos para guiar el proceso de toma de decisiones.

Las IA comunes en equipos personales y smartphones incluyen a Siri, Cortana y el Asistente de Google. Tal como se muestra en la figura 17.1, estos recursos de IA le permiten comunicarse, a menudo mediante comandos de voz, para preguntar por direcciones, establecer recordatorios, buscar en la web y más.

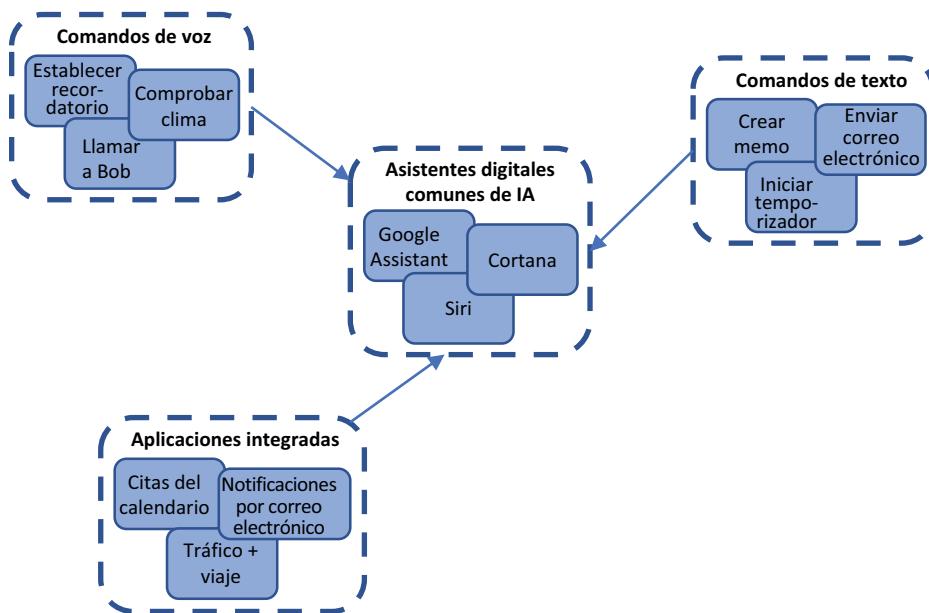


Figura 17.1 Un uso común de la IA en la vida diaria es la que se realiza en asistentes digitales como Cortana, Siri y el Asistente de Google. Puede usar comandos de voz o texto para interactuar con ellos, y estos pueden monitorear las condiciones diarias del calendario y del traslado al trabajo para así advertirle de los problemas de tráfico.

Los asistentes digitales como estos normalmente no implican el uso de grandes cantidades de lo que pudiera considerar *inteligencia*. Estos escuchan y responden a la entrada que usted proporciona. Sin embargo, esas entradas pueden variar y no siempre ser comandos específicos. Piense en cómo un asistente digital le permite establecer un recordatorio. Puede usar una de las siguientes frases:

- “Recuérdame pasar a comprar leche a las 5”.
- “Dime que debo pasar a comprar leche de camino a casa”.
- “Necesito comprar leche cuando esté en la tienda”.

Si desarrolló una aplicación tradicional, deberá escribir código que pueda manipular todas las posibles variaciones de la forma en la que un usuario podría proporcionar instrucciones. Podría crear expresiones de uso habitual para ayudar a capturar algunas de las variaciones, ¿pero qué ocurre cuando el usuario dice una frase que usted no programó? ¿O si la interacción se realiza vía texto y la solicitud tiene un error de ortografía que no previó? Estos tipos de interacciones son excelentes para la IA. Tal como se muestra en la figura 17.2, la aplicación se programa con varias frases comunes y luego se puede hacer una conjectura fundada basándose en lo que se “piensa” que el usuario está pidiendo.

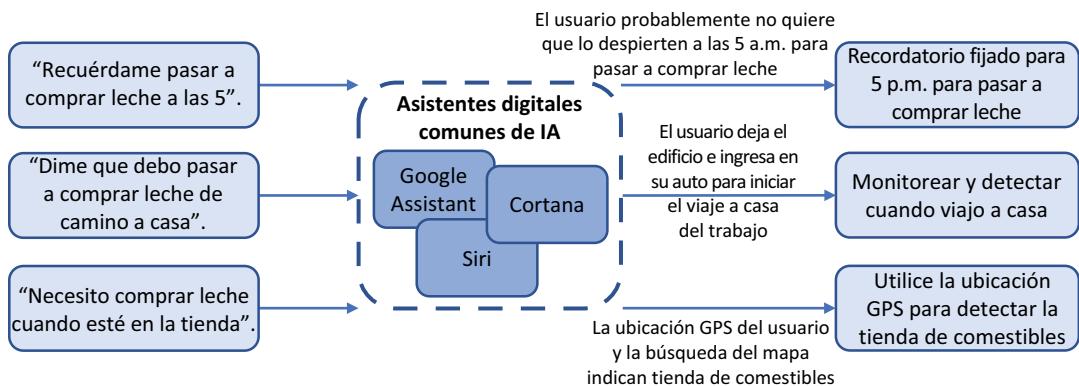


Figura 17.2 La IA puede tomar la entrada del usuario y tomar las decisiones que mejor se adapten a la acción prevista. La IA no está preprogramada con todas estas respuestas y árboles de decisión posibles. En cambio, usa modelos de datos y algoritmos que aplican contexto a la entrada del usuario e interpretan el significado y el resultado apropiado.

(Aún) no es verdadera inteligencia, incluso en formas complejas de IA; en cambio, es una conjectura fundada que se basa en un modelo de datos con el que se ha entrenado a la IA. Este modelo de datos puede incluir muchas variaciones y frases, y puede ser capaz de aprender nuevos significados con el tiempo. ¿Cómo se aprende y de dónde provienen estos modelos de datos? Ahí es donde ML se vuelve importante.

17.1.2 Machine learning

Una de las grandes palabras de moda en la informática durante los últimos años ha sido *macrodatos*. El concepto es que los sistemas informáticos, en especial en la nube, son un gran recurso para procesar grandes cantidades de datos. Cantidades *muy* grandes de datos. Estos trabajos de procesamiento pueden ejecutarse por unos minutos o por horas, según la cantidad de los datos y los cálculos que se requieren, además de permitirle preparar y analizar grandes volúmenes de datos para determinar patrones y correlaciones específicos. Estos aprendizajes forman modelos de datos que otras aplicaciones o

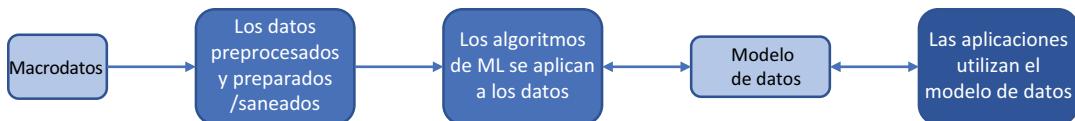


Figura 17.3 Grandes cantidades de datos sin procesar se procesan y preparan para su uso. Se pueden aplicar diferentes técnicas de preparación y saneamiento de datos, en función de las entradas sin procesar. Entonces, los algoritmos de ML se aplican a los datos preparados para desarrollar un modelo de datos apropiado que refleje la mejor correlación entre todos los puntos de datos. Con el tiempo, pueden producirse y refinarse diferentes modelos de datos. Las aplicaciones pueden usar los modelos de datos en sus propias entradas de datos para guiar sus patrones de toma de decisiones y comprensión.

IA pueden usar como ayuda para tomar decisiones. Tal como se muestra en la figura 17.3, ML implica algunos pasos e incluye entradas y salidas.

Así es como funciona la forma más básica de ML:

- 1 Para comenzar con el proceso, se proporcionan grandes cantidades de datos sin procesar como entrada.
- 2 Estos datos se procesan y preparan en un formato práctico que permite enfocarse en los puntos de datos específicos que se requieren para el análisis.
- 3 Los algoritmos de ML se aplican a los datos. Aquí es donde se produce el verdadero cálculo numérico. Los algoritmos están diseñados para detectar y procesar similitudes o diferencias en el gran número de puntos de datos.
- 4 Basándose en el análisis de los algoritmos, se produce un modelo de datos que define patrones dentro de los datos. Estos modelos de datos pueden refinarse con el tiempo si partes del modelo resultan ser incorrectas o estar incompletas cuando se aplican datos adicionales del mundo real.
- 5 Las aplicaciones usan los modelos de datos para procesar sus propios conjuntos de datos. Estos conjuntos de datos normalmente son mucho más pequeños que los datos sin procesar que se proporcionan a los algoritmos de ML. Si el modelo de datos es válido, entonces incluso con una pequeña entrada de datos de la aplicación, puede determinarse el resultado o la correlación correcta.

ML a menudo implica algoritmos complejos diseñados para procesar todos los puntos de datos proporcionados. Hadoop y Apache Spark son dos pilas de aplicaciones comunes que se usan para procesar macrodatos. Azure HDInsight es un servicio administrado que le permite analizar los grandes conjuntos de datos procesados por estas pilas de aplicaciones. Para profundizar un poco más en el análisis y los algoritmos, los científicos de datos utilizan comúnmente el lenguaje de programación R para ayudar a desarrollar los modelos requeridos. No se preocupe demasiado por saber qué es Hadoop o R. El punto clave es que Azure puede ejecutar las herramientas comunes de ML que se aceptan ampliamente en la industria.

17.1.3 La unión de IA y ML

Una aplicación común en un smartphone es la aplicación de navegación, tal como se muestra en la figura 17.4. Su proveedor, por ejemplo Google, puede rastrear la ruta que toma cada día para ir al trabajo, a qué hora sale normalmente de casa y cuánto tarda en llegar.

En este ejemplo de Google Maps se muestra cómo la IA y ML trabajan juntos. La IA se aplica para saber cuándo generar una notificación basada en los datos recibidos después del procesamiento del modelo de datos de ML. Otro ejemplo del trabajo

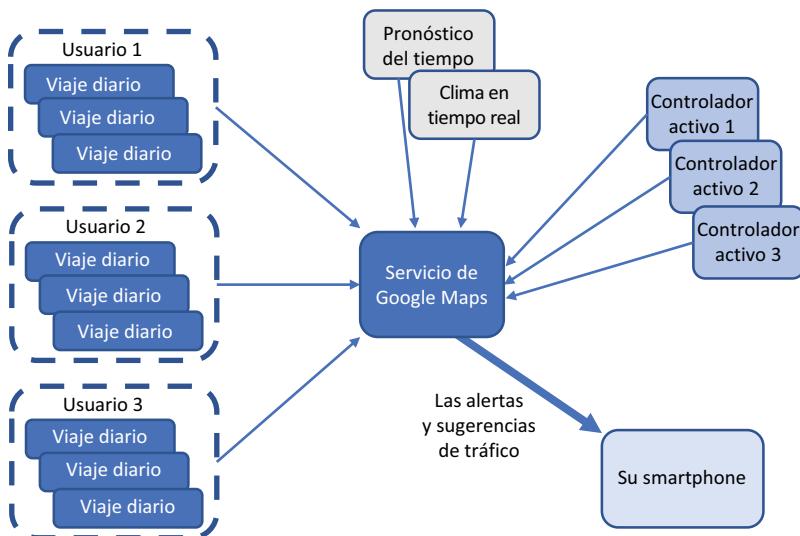


Figura 17.4 Cada día, el servicio Google Maps recibe múltiples puntos de datos de usuarios, los que registran los detalles de su desplazamiento al trabajo. Estos datos pueden prepararse y procesarse, junto con el pronóstico del tiempo y las condiciones climáticas en tiempo real durante estos trayectos. Los algoritmos de ML se pueden aplicar a estos grandes conjuntos de datos. Además, se puede producir un modelo de datos. Como una muestra más pequeña de conductores activos ingresa sus condiciones de viaje actuales o los datos del tiempo en el servicio Google Maps, el modelo de datos puede aplicarse para predecir su trayecto y generar una alerta de tráfico que se envía a sus smartphones y sugiere una ruta a casa alternativa.

conjunto entre IA y ML es la idea previa de establecer un recordatorio para comprar leche. Si la IA se entrenara con modelos de datos de ML, el asistente sabría que probablemente compra leche en la tienda de abarrotes, por lo que no se lo recordaría si va a la ferretería. El modelo de datos de ML también podría ayudar a la IA a entender que existe una mayor probabilidad de que usted desee recibir un recordatorio a las 5:00 p.m., no a las 5:00 a.m., por lo que no lo despertaría a las 5:00 a.m. para que vaya a comprar leche. Si su smartphone rastrea que sube a su automóvil a las 5:00 p.m. y se marcha del trabajo, ML generará un modelo de datos que predice que va camino a casa, por lo que esa hora será un buen momento para que la IA le recuerde que debe comprar leche.

Estos ejemplos básicos, pero poderosos, muestran cómo se usa ML para mejorar la IA. Usted entrena la IA proporcionando un conjunto de puntos de datos que ML procesa para mejorar la precisión o la toma de decisiones.

17.1.4 Herramientas de ML de Azure para científicos de datos

Quiero hablar rápidamente de algunas formas en las que se puede hacer el trabajo de cálculos numéricos y ML en el mundo real. Para que todos puedan acceder a este capítulo, los ejercicios usan Microsoft Bot Framework para IA, y ML con Language Understanding Intelligent Service (LUIS). Para hacer el trabajo sucio con ML, debemos enfocarnos un poco más en el procesamiento de datos y los algoritmos.

En Azure, un par de componentes geniales le ayudarán a profundizar en los datos a escala masiva. El primero es Azure Machine Learning, un servicio basado en la web que le permite desarrollar experiencias visualmente mediante la incorporación de conjuntos de datos y modelos de análisis. Estos experimentos pueden usar orígenes de datos como Hadoop y SQL, además del soporte de programación adicional que proporcionan lenguajes como R y Python. Puede arrastrar y soltar los orígenes de datos, las técnicas de preparación de datos y los algoritmos de ML. Puede ajustar esos algoritmos y luego revisar y ajustar los modelos de datos producidos.

Azure Machine Learning proporciona un obstáculo bajo para ingresar a los recursos informáticos de gran escala disponibles en Azure. Un beneficio principal de la realización de un cálculo de datos de ML en Azure es que puede tener acceso a una gran cantidad de potencia de proceso y usarlo solo durante el tiempo necesario para realizar los cálculos. En los entornos tradicionales, esos costosos recursos informáticos se mantendrían inactivos por períodos prolongados entre trabajos de procesamiento de datos.

Otro recurso genial que le ayuda a realizar ML y cálculos numéricos serios en Azure son las máquinas virtuales de ciencia de datos (DSVM). Estas VM están disponibles para Linux y Windows. Vienen con muchas aplicaciones comunes preinstaladas, como Jupyter Notebooks, Anaconda Python y R Server o SQL Server (figura 17.5).



Figura 17.5 Las DSVM están disponibles para Windows y Linux. Esta DSVM de Windows Server 2016 viene con varias aplicaciones de ciencia de datos preinstaladas, tales como R Server y Jupyter Notebooks. Las DSVM le permiten ponerse en marcha rápidamente con el procesamiento de macrodatos y el desarrollo de algoritmos de ML.

No es necesario instalar todas las herramientas y dependencias en su equipo local, puede crear una DSVM con tantos recursos de CPU y memoria como necesite para procesar rápidamente sus datos, y luego eliminar la VM cuando se haya completado el trabajo de procesamiento y tenga los modelos de datos que necesita.

17.2 Azure Cognitive Services

Bien, ¿qué hay de los servicios de IA para hacer sus aplicaciones más inteligentes? En Azure, un conjunto de servicios relacionados componen el conjunto de aplicaciones de Cognitive Services. Los servicios abarcan algunas áreas comunes de la IA que le permiten integrar rápidamente estos recursos inteligentes en sus aplicaciones, divididas en las siguientes áreas generales:

- Visión
- Voz
- Lenguaje
- Decisión
- Búsqueda

Más de dos docenas de servicios forman parte de la familia de Cognitive Services. Algunos de ellos son

- *Visión*, que incluye
 - *Computer Vision* para análisis de imágenes, subtítulos y etiquetado.
 - *Face* para analizar y detectar rostros en imágenes.
- *Voz*, que incluye
 - *Speech Services* para analizar y convertir la voz en texto, y viceversa.
 - *Speaker Recognition* para identificar y verificar al orador.
- *Lenguaje*, que incluye
 - *Language Understanding (LUIS)* para comprender y procesar la interacción con los usuarios. Exploraremos LUIS en el laboratorio al final de este capítulo.
 - *Translator Text* para analizar y corregir errores ortográficos o realizar traducciones.
- *Decisión*, que incluye
 - *Content Moderator* para revisar y moderar fotos, videos y texto.
 - *Personalizer* para analizar patrones y ofrecer recomendaciones a los clientes.
- *Búsqueda*, que incluye
 - *Bing Custom Search* para implementar la búsqueda en sus datos personalizados y dentro de las aplicaciones.
 - *Bing Autosuggest* para proporcionar sugerencias automáticas cuando los usuarios ingresan frases de búsqueda y consultas.

Como puede ver, muchos servicios de Azure combinan características de IA y ML. Este capítulo se enfoca en el lenguaje, específicamente en LUIS. Este servicio se usa comúnmente para crear un bot inteligente que pueda ayudar a los clientes en su sitio web. A continuación, puede desarrollar una aplicación que use los servicios de IA en Azure que pueda interpretar frases y preguntas, además de proporcionar la respuesta apropiada para guiar a un usuario a través de un proceso de pedido o una solicitud de soporte.

17.3 Creación de un bot inteligente para ayudar con pedidos de pizza

Un *bot* es una aplicación que está programada para responder a las tareas y las entradas de un usuario. Si esto suena muy similar a cualquier aplicación normal, es porque bueno, es bastante similar. La diferencia está en cómo la aplicación de bot determina la respuesta.

Un bot básico y común a menudo no es más que una aplicación que ofrece alguna forma de automatización. Cuando un usuario envía un mensaje, establece una etiqueta en un mensaje de correo electrónico o envía un término para búsqueda, el bot lleva a cabo tareas programadas que realizan una acción específica. Aquí no hay una verdadera IA o ML; la aplicación del bot solo responde a la entrada del usuario.

Con el marco correcto, un bot puede ampliarse y se le puede conceder un poco más de libertad e inteligencia. Al comienzo de nuestra descripción general de IA, analicé cómo una aplicación típica debe preprogramarse con todas las entradas de usuario previstas y cuál sería la salida correspondiente. Sin embargo, si por ejemplo el usuario proporciona una frase de entrada diferente o comete un error ortográfico, no hay flexibilidad.

Microsoft produce el Bot Framework, que permite a un bot de Azure integrar fácilmente los SDK de Bot Builder y conectarlos a Azure Cognitive Services. Con una experiencia mínima en códigos, puede crear bots inteligentes que usen el poder de Azure para entregar una gran experiencia al cliente. ¡No intente crear Skynet a menos que haya visto el final de *Terminator*!

17.3.1 Creación de un bot de Azure Web App

Vamos a implementar un bot e integrarlo con algunos servicios de IA y ML. El bot se ejecuta en Azure Web App y usa Microsoft Bot Framework para conectarse a LUIS y permitir que un cliente pida una pizza. En la figura 17.6 se describe lo que estos ejercicios crearán y los servicios que se usan.

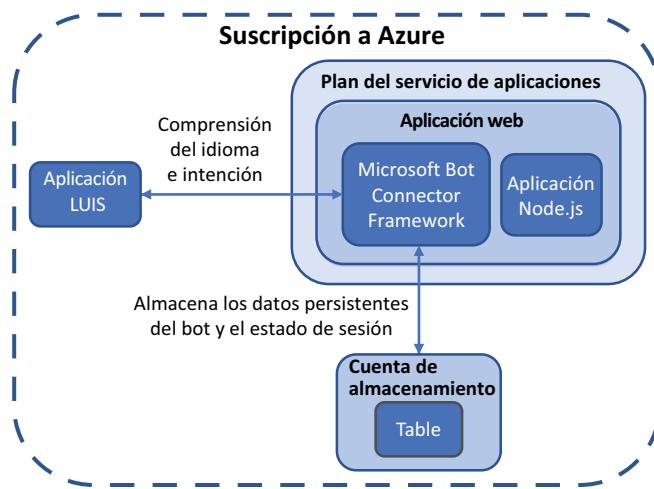


Figura 17.6 En los siguientes ejercicios, creará un bot de aplicación web que integre múltiples servicios de IA y ML de Azure para interactuar con un cliente y ayudarlo a pedir una pizza.

Pruébelo ahora

Para crear un bot de aplicación web de Azure, complete los pasos siguientes:

- 1 Abra Azure Portal y seleccione Crear un recurso en la esquina superior izquierda.
- 2 Busque y seleccione Bot de Web App y, a continuación, seleccione Crear.
- 3 Escriba un nombre para su bot, como `azuremol`; luego, cree un nuevo grupo de recursos y dele un nombre, como `azuremolchapter17`.
- 4 Seleccione la región más apropiada para usted y elija el nivel de precios F0. El bot no procesará muchos mensajes, por lo que el nivel gratuito (F0) está bien.
- 5 Seleccione una plantilla de bot y elija el lenguaje de SDK de Node.js.
- 6 Cree un bot básico, ya que proporcionaremos nuestro propio código de aplicación de ejemplo en un ejercicio posterior. Este paso crea una aplicación LUIS que puede utilizar para realizar entrenamiento de idiomas y ML.
- 7 Elija la región más adecuada para su aplicación LUIS y cree una nueva cuenta de LUIS.
- 8 Proporcione un nombre para la cuenta de LUIS, como `azuremol`. Esta cuenta de LUIS se encarga de la opinión del usuario para nuestro bot.
- 9 Elija Plan de servicio de la aplicación y cree un nuevo plan. Proporcione un nombre, como `azuremol`, y nuevamente, seleccione la región más apropiada para usted.
- 10 Desactive Información de la aplicación, porque su bot no lo usará. Al igual que con los capítulos anteriores sobre aplicaciones web, para el uso de producción le recomendamos aprovechar el poder de App Insights para obtener visibilidad en el rendimiento de su aplicación mediante la transmisión de datos y análisis directamente desde el código.
- 11 Acepte la opción para crear automáticamente la ID y contraseña de la aplicación de Microsoft, acepte el acuerdo y elija Crear.

Tardará unos minutos en crear el bot de aplicación web y sus componentes asociados. Ocurren muchas cosas en segundo plano:

- Se crea un plan de Azure App Service.
- Se implementa una aplicación web, junto con una aplicación web Node.js de ejemplo.
- Se crea una aplicación LUIS y las claves de conexión se configuran con su aplicación web.
- Se crea un bot con Microsoft Bot Connectory las claves de conexión se configuran desde su aplicación web.

17.3.2 Lenguaje e intención de comprensión con LUIS

Una de las áreas de Azure Cognitive Service que observamos anteriormente es el lenguaje. Esto tiene sentido, porque a menudo se usa alguna forma de lenguaje para interactuar con una IA. Puede usar LUIS para procesar un mensaje o una frase del usuario y determinar su intención. Esa intención ayuda a que su aplicación proporcione una respuesta apropiada. Vamos a ampliar su bot con LUIS.

Pruébelo ahora

Para crear una aplicación LUIS y usar ML para entrenarla, complete los pasos siguientes

- 1 Abra un navegador web para www.luis.ai, e inicie sesión con las mismas credenciales de Microsoft que su suscripción de Azure.
- 2 Seleccione Ir a mis aplicaciones y elija su aplicación, como `azuremol`. El nombre de su aplicación LUIS probablemente tendrá algunos caracteres numéricos anexados desde el nombre de bot que especificó en Azure Portal.
Se crearon algunas intenciones preconfiguradas, sin embargo, le recomendamos sobreescribir la aplicación LUIS con un ejemplo más enfocado en la pizzería.
- 3 Descargue el archivo `azuremol.json` desde GitHub en <https://github.com/fouldsy/azure-mol-samples-2nd-ed/blob/master/17/luisapp/azuremol.json> a su equipo local. Para facilitar las cosas, seleccione el botón Sin procesar en GitHub para ver solo los contenidos del archivo.
- 4 De regreso en su aplicación LUIS, elija Administrar la aplicación, y luego seleccione Versiones.
- 5 Elija importar una versión, examine y seleccione el archivo `azuremol.json` que descargó, introduzca el nombre de versión 1.0 y luego seleccione Listo.
- 6 Regrese a Crear en el menú superior para ver las intenciones importadas de la aplicación de ejemplo. Elija una o dos de las intenciones, tales como `greetings` u `orderFood`, y observe algunas de las frases de ejemplo que un cliente podría usar para comunicarse con el bot.
- 7 Antes de que pueda ver la aplicación en acción, debe entrenarla. Seleccione Entrenar y espere unos segundos para que se complete el proceso. En la figura 17.7 se muestran los procesos de ML en funcionamiento para entrenar a su aplicación LUIS.

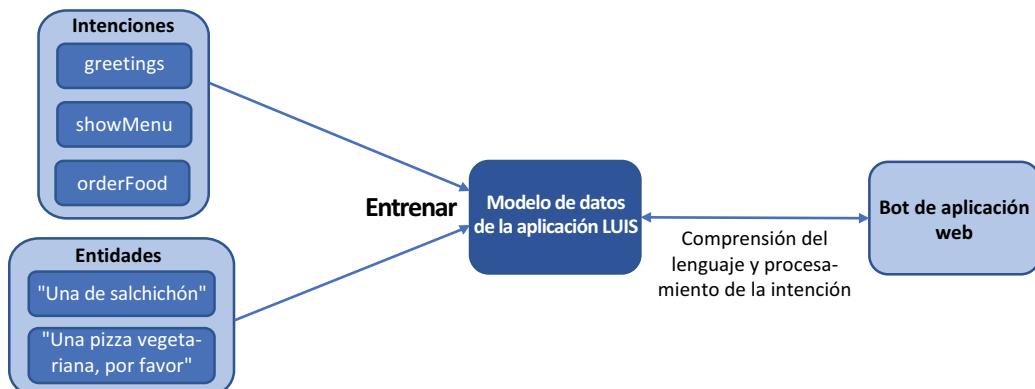


Figura 17.7 Cuando se entrena la aplicación LUIS, las intenciones y entidades se ingresan y procesan para crear un modelo de datos. Entonces, el bot de aplicación web usa este modelo de datos para procesar la comprensión e intención del lenguaje. La cantidad de intenciones y entidades que se ingresan para procesamiento es pequeña, por lo que el modelo de datos no es perfecto. En el mundo real, se proporcionarían muchas más intenciones y entidades, y usted entrenaría, probaría y refinaría repetidamente el modelo de datos para crear progresivamente conjuntos de datos más grandes que desarrollen un modelo exacto de procesamiento del lenguaje y las intenciones.

En una aplicación real, más compleja, este proceso de entrenamiento podría tardar más en completarse, ya que todas sus intenciones y entidades serán procesadas por los algoritmos de ML para desarrollar el modelo de datos necesario para que su aplicación responda de forma apropiada a la comunicación con el cliente.

- 8 Con la aplicación LUIS entrenada, seleccione Probar y escriba un par de saludos, como *hola*. Bajo cada uno de sus mensajes se encuentra la intención con la mejor puntuación, junto con la probabilidad de que el mensaje o la expresión que ingresó coincida con la intención. Estos saludos básicos deben coincidir correctamente con la intención greetings.
- 9 Intente ingresar otro saludo, como *(buenas) tardes* o *(buenas) noches*. Los saludos de una palabra basados en el momento del día pueden devolver una intención con la mejor puntuación incorrecta, como orderStatus. Pruebe algunas otras frases hasta que estas no coincidan con la intención esperada, lo que indica que la aplicación LUIS no entiende completamente lo que dice. Seleccione uno de sus mensajes incorrectos, como *días* y elija Inspeccionar.
- 10 En el menú Inspeccionar, elija editar la intención con la mejor puntuación incorrecta. En el menú desplegable, elija greetings o la intención más apropiada para la frase incorrecta.
- 11 Ha hecho un cambio en la aplicación, así que vuelva a elegir Entrenar la aplicación LUIS. En la figura 17.8 se muestra cómo proporcionar entradas adicionales para que los algoritmos de ML procesen el modelo de datos y refinen la comprensión e intención del lenguaje.
- 12 En la ventana de mensajes de prueba, ingrese nuevamente el mensaje incorrecto, como *días*. Esta vez, el intento con la mejor puntuación debiera identificarlo como greetings.
- 13 Para hacer que la aplicación LUIS actualizada esté disponible para el bot de aplicación web, elija la opción Publicar en el menú superior. Acepte todos los valores predeterminados y elija publicar en una ranura de producción. El proceso de publicación tardará unos segundos en completarse.

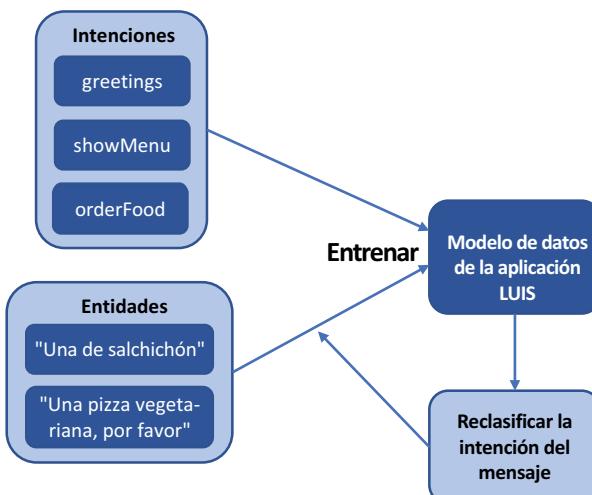


Figura 17.8 Cuando reclasifica la intención de los mensajes y vuelve a entrenar la aplicación LUIS, el modelo de datos se refina a medida que se proporcionan entradas de datos adicionales a los algoritmos de ML. Cuando ingrese saludos similares en el futuro, el modelo de datos (idejalmente) habrá mejorado y responderá de forma más apropiada.

Recuerde, el bot se ejecuta en una aplicación web, tal como las ranuras de producción y los espacios de ensayo que conoció en el capítulo 3. En el mundo real, debe publicar en un espacio de ensayo, verificar que todo funcione según lo esperado y luego publicar en la ranura de producción. Las mismas características de PaaS que le permitieron probar y mover el código web entre los ciclos de vida de implementación y producción también benefician al ciclo de vida de su bot de aplicación web con tecnología de LUIS.

En este ejemplo básico, ML pudo reconocer la entrada de datos de *(buenos) días* como un saludo y entender que saludos similares como *(buenas) noches*, también son saludos. ML funciona mejor cuando se puede ingresar un conjunto grande de datos al modelo de datos, por lo que es importante probar exhaustivamente y ayudar a entrenar su aplicación. La IA, en este caso la aplicación LUIS, solo es tan buena como el tamaño y la calidad de los datos proporcionados a los algoritmos de ML.

17.3.3 Creación y ejecución de un bot de aplicación web con LUIS

Ahora tiene un bot de aplicación web en Azure y una aplicación LUIS que manipula el procesamiento del lenguaje y devuelve la intención del cliente. Para integrar ambos, se debe modificar el código del bot para usar LUIS. Los SDK están disponibles para los lenguajes de programación C# y Node.js. Creo que Node.js hace que sea un poco más rápido y fácil entender lo que ocurre en el código, si esto es nuevo para usted. Si está familiarizado con C#, está invitado a explorar el SDK de C# cuando haya terminado este capítulo. Por ahora, usemos una aplicación básica de Node.js del repositorio de ejemplo GitHub para ver el bot en acción con LUIS.

Pruébelo ahora

Para actualizar su bot de aplicación web con el bot de LUIS entrenado, complete los pasos siguientes:

- 1 En Azure Portal, seleccione Grupos de recursos en el menú de la izquierda y elija su grupo de recursos, como `azuremolchapter17`; a continuación, seleccione su bot de aplicación web, como `azuremol`.

Utilicemos un bot de ejemplo de nuestro repositorio de ejemplos de GitHub. El bot de ejemplo está escrito en Node.js, pero al igual que con las aplicaciones de ejemplo anteriores, no se preocupe si eso no es lo suyo.

- 2 Para implementar el bot de ejemplo, abra Cloud Shell. Si fuera necesario, clone el repositorio de ejemplos de GitHub en su Cloud Shell de la siguiente manera:

```
git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
```

- 3 Cambie al directorio para iniciar el capítulo 17:

```
cd azure-mol-samples-2nd-ed/17/webapppbot
```

- 4 Inicialice el repositorio Git y agregue los archivos de bot:

```
git init && git add . && git commit -m "Pizza"
```

- 5 Para cargar el bot de ejemplo, cree una conexión a su aplicación web. El siguiente comando obtiene el repositorio de la aplicación web y configura su repositorio Git de ejemplo local para conectarse con él. En capítulos anteriores, tuvo que buscar esta dirección; pero por ahora espero que haya empezado a explorar qué más puede hacer la CLI de Azure y se haya dado cuenta de que gran parte de esta información se puede obtener rápidamente.

```
git remote add webappbot \
$(az webapp deployment source config-local-git \
--resource-group azuremolchapter17 \
--name azuremol \
--output tsv)
```

- 6 Inserte el bot Node.js de ejemplo en su aplicación web con el siguiente comando:
`git push webappbot master`
- 7 Cuando se le solicite, ingrese la contraseña para el usuario de Git que creó y utilizó en capítulos anteriores (la cuenta creada en el capítulo 3).

Si no escribió su contraseña de Git en una nota rápida

Si olvidó la contraseña, puede restablecerla. Primero, obtenga el nombre de usuario de su cuenta de implementación de Git local:

```
az webapp deployment user show --query publishingUserName
```

Para restablecer la contraseña, introduzca el nombre de su cuenta desde el comando anterior y, a continuación, siga las instrucciones para configurar una nueva contraseña. El siguiente ejemplo restablece la contraseña de la cuenta de usuario denominada `azuremol`:

```
az webapp deployment user set --user-name azuremol
```

Demos un vistazo a la figura 17.9 para ver lo que ha implementado. La aplicación LUIS ahora está entrenada con algoritmos de ML y su modelo de datos está listo para que la aplicación Node.js permita a los clientes interactuar y pedir pizza.

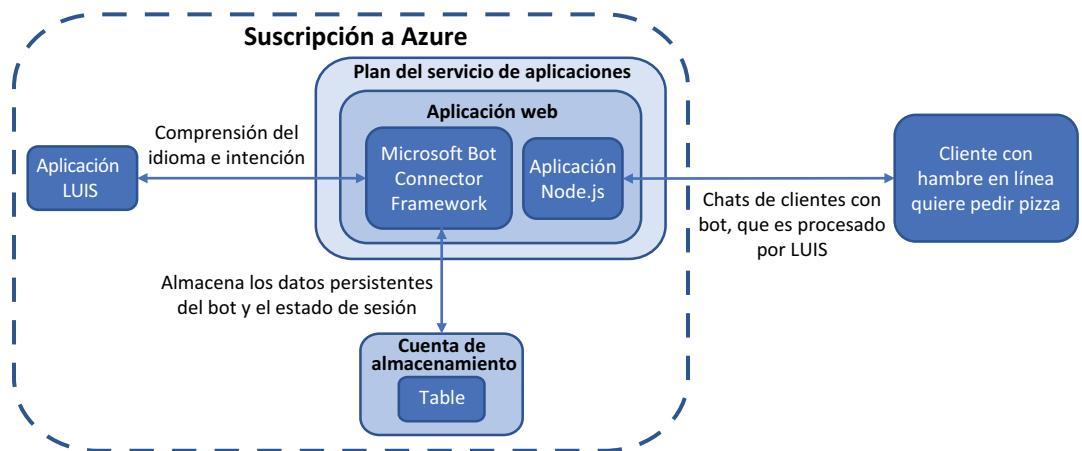


Figura 17.9 Un cliente ahora puede tener acceso a su bot en línea y solicitar el menú o pedir pizza. LUIS proporciona la comprensión del lenguaje, lo que permite que el bot procese pedidos y los envíe a Azure Storage para procesamiento adicional.

De regreso en Azure Portal para su bot de aplicación web, seleccione Probar en chat web. La primera vez que se conecta al bot este tarda unos segundos, pero luego debiera poder interactuar, ver la lista de pizzas en el menú y crear un pedido, tal como se muestra en la figura 17.10. ¡Pruébelo!

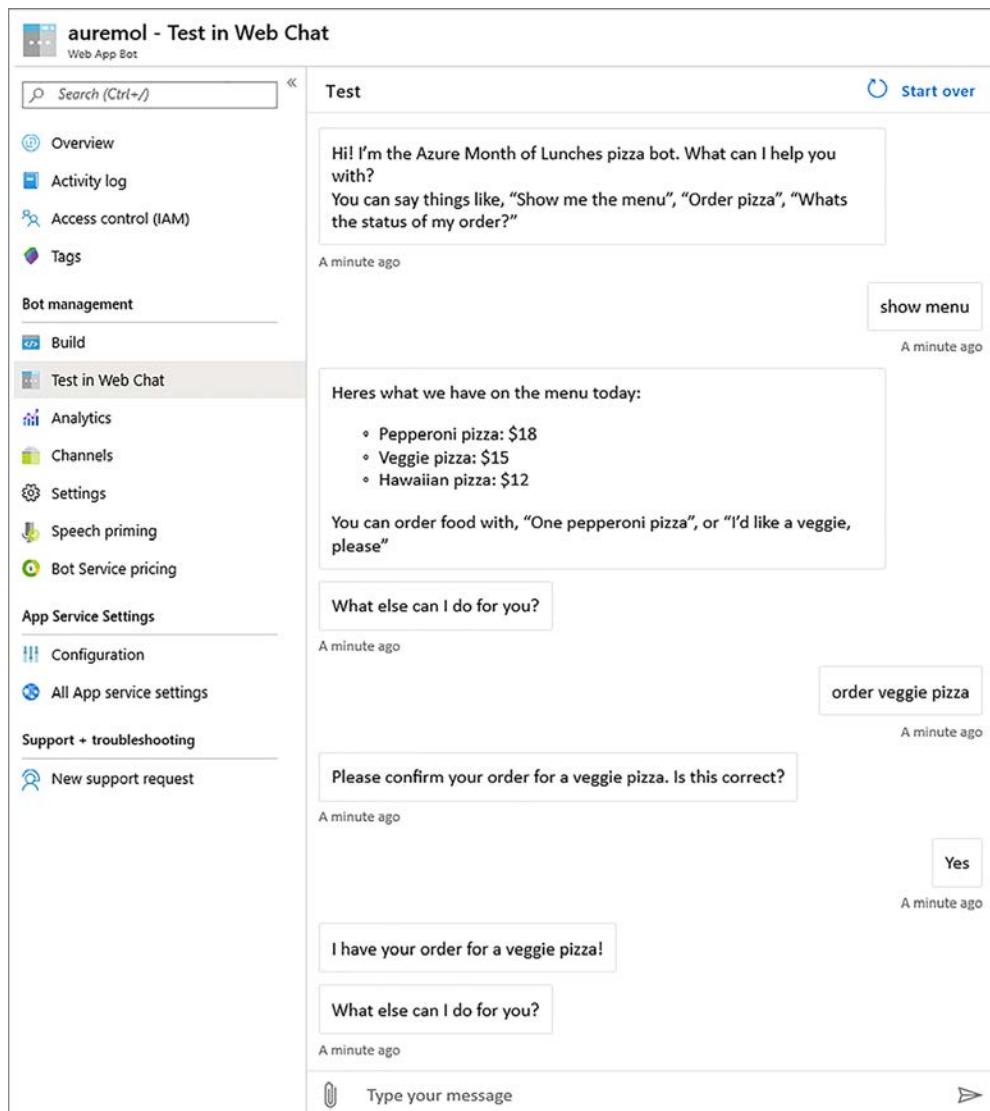


Figura 17.10 Con el bot de aplicación web en ejecución, inicie una conversación e intente pedir una pizza. En este cuadro de diálogo de ejemplo, puede ver el menú, pedir una pizza y revisar el estado del pedido. La aplicación es básica, en realidad no está creando pedidos ni actualizando el estado de la pizza que se pidió, sin embargo, espero que el ejercicio le muestre cómo puede implementar rápidamente un bot en Azure.

Espero que estos ejercicios básicos le hayan entregado una idea de lo que Azure puede ofrecer para IA y ML. El bot de aplicación web con LUIS puede expandirse para incluir Azure Cognitive Services adicionales como Spell Check y Translator. Estos servicios le permiten interpretar palabras y frases si el usuario las escribe incorrectamente, o permite que su bot converse en varios idiomas. O bien, podría usar Face API y Personalizer para detectar qué cliente hizo un pedido basándose en el reconocimiento facial de la cámara y automáticamente recomendar pizzas que podrían gustarle.

ML era parte de la aplicación LUIS, sin embargo hay muchos otros recursos y herramientas de ML disponibles en Azure. La capacidad de procesar grandes conjuntos de datos y modelos de datos de ML en recursos de procesamiento de Azure de alto rendimiento reduce la entrada para que usted pueda desarrollar aplicaciones respaldadas por algunos conjuntos de datos serios. Las aplicaciones son más exactas y eficientes, y no se debe comprar ningún hardware o herramientas especiales para instalarlas, ya que los DSVM incluyen todos los componentes necesarios. No todas las aplicaciones son adecuadas para IA y ML, sin embargo, a medida que los clientes esperan más características inteligentes con respecto a lo que su negocio puede ofrecer, estos servicios Azure a menudo pueden ayudarlo a diferenciarse.

Procesamiento de cargas de trabajo por lotes

Las otras dos áreas de Azure que podría ser de interés en términos de macrodatos e informática para ML son los servicios de Azure Batch y HPC. Azure Batch le permite realizar tareas de procesamiento grandes y repetitivas sin necesidad de administrar clústeres de programadores para el trabajo. Batch ejecuta tareas en VM con su propia administración y programador para ayudarlo, así como los conjuntos de escala incluyen autoescalamiento y equilibrio de cargas para VM. Aunque Batch no tiene relación directa con ML, si necesita otras tareas de procesamiento informático grandes, Batch es una excelente opción.

También existen en Azure componentes de informática de alto rendimiento (HPC) para VM de gran tamaño o acceso a VM de unidad de procesamiento gráfico (GPU). También se pueden usar herramientas y conjuntos de aplicaciones específicos como DataSynapse y Microsoft HPC Pack para ejecutar aplicaciones que demandan el uso de grandes cantidades de potencia de proceso.

Áreas como ML, Azure Batch y HPC son ejemplos excelentes de cómo usar proveedores de informática en la nube como Azure para ejecutar tareas de procesamiento de gran tamaño. Usted solo paga por los recursos informáticos que usa, por lo que no es necesario comprar y mantener equipos costosos para un uso mínimo.

17.4 Laboratorio: Cómo agregar canales para la comunicación de un bot

En los ejemplos anteriores, se comunicó con su bot a través de una ventana de prueba en Azure Portal. Los canales le permiten ampliar la forma en la que puede interactuar con el bot. Puede permitir que el bot se comunique con Skype o Facebook Messenger, o con aplicaciones como Microsoft Teams y Slack. Azure Bot Service simplifica los pasos necesarios para integrar un bot con esos servicios externos:

- 1 En Azure Portal, seleccione el bot de aplicación web y luego elija Canales.
- 2 Elija un canal que le guste, como Skype.

Otras canales a menudo requieren que cree una conexión de desarrollador, como Facebook o Slack. Skype le permite copiar y pegar algún código HTML para que funcione.

- 3 Proporcione la información que se le solicite como la ID de aplicación del bot. Podrá encontrar esta ID en Configuración para la administración del bot.
- 4 Si es necesario, use el editor de código en línea para crear una página HTML básica, como default.htm, en el directorio wwwroot, y pegue cualquier código incrustado para su canal. Puede abrir la aplicación web desde Azure Portal y luego seleccionar su URL para abrir la página default.htm que indica su código de canal, como <http://azuremol.azurewebsites.net/default.htm>.

Azure Automation

Cuando sea posible, no debería iniciar sesión manualmente en un servidor y hacer cambios. No es necesario instalar el software haciendo clic en los botones de una GUI. Además, no es necesario realizar las actualizaciones en los archivos de configuración de un editor de texto. Estas acciones manuales crean una oportunidad para que se produzcan errores, lo que puede tener como resultado configuraciones erróneas y errores de aplicación. Si desea replicar la configuración de un servidor, ¿puede recordar todos los pasos necesarios para poner en marcha el servidor existente? ¿Qué ocurre si debe hacerlo nuevamente en seis meses?

En el capítulo 16, nos referimos a una forma de comprobar y aplicar automáticamente las actualizaciones a los servidores. Esto se realizó con el uso de Azure Automation. En este capítulo, examinamos cómo puede crear, ejecutar y editar runbooks, además de usar Desired State Configuration de PowerShell para instalar aplicaciones y configurar servidores automáticamente.

18.1 ¿Qué es Azure Automation?

Una cuenta de Azure Automation reúne muchos elementos, tal como se muestra en la figura 18.1. Una característica principal es la de crear y ejecutar scripts a petición o en un programa definido. Puede crear scripts en PowerShell o Python, y dejar que la plataforma Azure manipule la programación y ejecución de esos runbooks. Puede compartir credenciales y objetos de conexión, y aplicar e informar automáticamente las configuraciones de servidores deseadas. Update Management, que analizamos en el capítulo 16, mantiene sus servidores seguros y actualizados con los parches y las actualizaciones de host más recientes a lo largo del ciclo de vida del entorno de su aplicación.

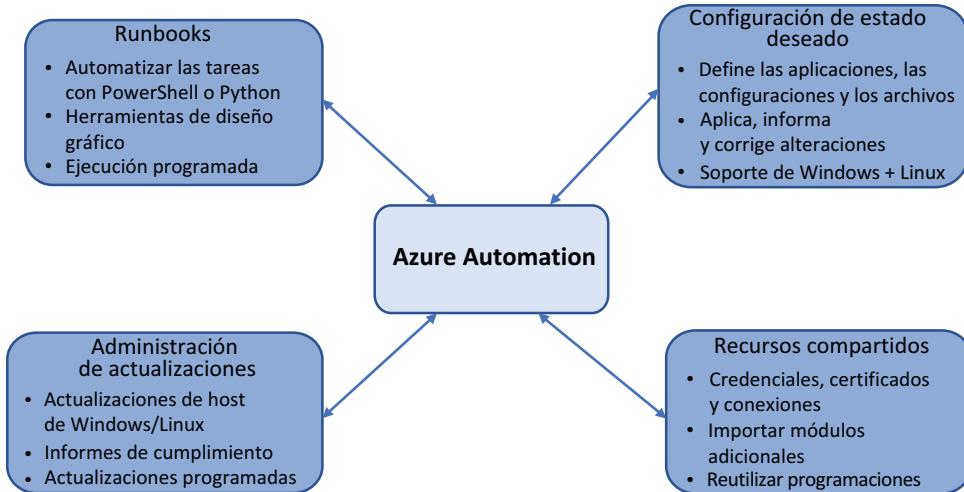


Figura 18.1 Azure Automation proporciona muchas características relacionadas. Un conjunto compartido de recursos, tales como credenciales, certificados, programas y objetos de conexión pueden utilizarse para ejecutar automáticamente scripts de PowerShell o Python en servidores de destino. Usted puede definir el estado deseado de un servidor y Azure Automation instala y configura el servidor de forma apropiada. Las actualizaciones de host y los parches de seguridad se pueden aplicar automáticamente. Todas estas características funcionan en servidores de Windows y Linux, en Azure y en entornos locales u otros proveedores de nube.

Para simplificar la administración en múltiples runbooks o configuraciones de estado deseado en una cuenta de Automation, puede compartir los siguientes recursos:

- *Los programas* le permiten definir un conjunto de tiempos y recurrencias que se puede aplicar a cada runbook o tarea de Update Management. Si posteriormente desea realizar el cambio a una aparición habitual, puede cambiar uno de los programas compartidos en lugar de cambiar cada runbook o tarea de Update Management individual que la use.
- *Los módulos* amplían la funcionalidad principal al almacenar módulos de PowerShell adicionales. Los módulos básicos Windows PowerShell y Azure ya están disponibles, pero los módulos adicionales, como aquellos para la administración de Linux, se pueden agregar y usar en runbooks.
- *Las credenciales* para las diferentes cuentas que tienen permisos para ejecutar varios runbooks se almacenan como activos, no se definen en cada runbook. Este enfoque le permite actualizar y restablecer las credenciales según sea necesario, y cada runbook que las usa se actualiza automáticamente. Por lo tanto, las credenciales no se almacenan como texto sin formato en runbooks, lo que aumenta la seguridad de los runbooks.
- *Las conexiones* definen las propiedades de autenticación para las entidades del servicio Azure AD. Este es un tipo especial de cuenta de usuario que permite a los runbooks tener acceso a sus recursos de Azure. Estas conexiones normalmente usan certificados digitales, no usan nombres de usuario y contraseñas, para proporcionar un nivel de seguridad adicional.

- Los *certificados* a menudo se integran con activos de conexión para proporcionar una forma segura de verificar la identidad de una entidad del servicio. Al igual que con las credenciales básicas, puede actualizar habitualmente estos certificados en una ubicación central, y cada runbook que los usa puede tener acceso de manera automática a los certificados nuevos. Puede crear y almacenar sus propios certificados para uso con runbooks o las definiciones de configuración de estado deseado.
- Las *variables* proporcionan un lugar central para el almacenamiento de valores de tiempo de ejecución como nombres, cadenas de ubicación y enteros. Cuando se ejecutan los runbooks, se inyectan estas variables. Este enfoque limita la cantidad de recursos codificados en cada runbook.

Trabajar más inteligente, no más duro

En el capítulo 16, nos referimos a cómo los servicios de administración de Azure funcionan en conjunto para supervisar e informar sobre los servidores en Azure, en entornos locales o en otros proveedores de nube. Los agentes requeridos se instalan y configuran en servidores remotos y luego proporcionan una forma de volver a conectarse a la infraestructura de Azure.

Azure Automation también puede funcionar a través de plataformas e infraestructuras. Por ejemplo, el trabajador de runbook híbrido puede ejecutar runbooks de Automation en servidores que se encuentran fuera de Azure. Usted continúa usando los activos compartidos de Automation que definen credenciales, conexiones y certificados, solo que esta vez esos activos se pueden usar para definir los componentes de autenticación para las diferentes plataformas. También puede usar las configuraciones de estado deseado en VM que no son de Azure, tanto para Windows como para Linux.

En todos los casos, se instala un componente de gateway en el entorno remoto que actuar como proxy de los comandos de Automation cuando se envían a los destinos designados. Este enfoque de proxy de gateway proporciona un punto de conexión único para Automation en los entornos remotos y minimiza las inquietudes de seguridad, ya que de otro modo no existe acceso directo a los servidores remotos.

Es posible que los runbooks y las definiciones de configuración de estado deseado deban editarse ligeramente para ejecutarse en servidores físicos locales en comparación con las VM de Azure. Al igual que con Azure Backup, Site Recovery o Update Management, la ventaja de Azure Automation es que proporciona un plano de administración único y un conjunto de herramientas para ofrecer automatización en todas sus diferentes infraestructuras y servidores.

18.1.1 Creación de una cuenta de Azure Automation

Vamos a dar el salto para crear una cuenta de Azure Automation y observaremos los runbooks predeterminados que se incluyen. Los runbooks de demostración ofrecen un marco excelente de creación de sus propios runbooks, además de contar con un editor gráfico que puede usar para arrastrar y soltar bloques de construcción que generan scripts de automatización.

Pruébelo ahora

Para crear una cuenta de Azure Automation y runbooks de ejemplo, complete los pasos siguientes:

- 1 En Azure Portal, seleccione Crear un recurso en la esquina superior izquierda.
- 2 Busque y seleccione Automatización y, a continuación, seleccione Crear.
La opción Automatización y control también crea un espacio de trabajo de Operations Management Suite (OMS) y configura Automation Hybrid Worker para administrar recursos fuera de Azure. OMS está en cierto modo en vías de extinción, reemplazado por los servicios básicos de Azure que vimos en capítulos anteriores. Por ahora, elija crear solo el recurso de automatización.
- 3 Escriba un nombre, como azuremol, y luego cree un nuevo grupo de recursos, como azuremolchapter18.
- 4 Seleccione la región de Azure más apropiada y cercana a usted, y acepte la opción Crear cuenta de ejecución de Azure.

La opción Crear ejecución como cuenta crea cuentas adicionales en Azure AD. También se crean certificados de seguridad que permiten la autenticación automática de las cuentas, sin necesidad de mensajes de usuario o de guardar una contraseña. Podría crear y especificar credenciales de cuentas normales adicionales, las que se definen como activo de Automation, para proporcionar un control más detallado de las cuentas que se usan para ejecutar ciertos runbooks.

Al combinarse con RBAC, que analizamos en el capítulo 6, es posible crear cuentas de ejecución para runbooks que ofrecen un conjunto limitado de permisos necesarios para lograr las tareas que requiere cada runbook o conjunto de runbooks. Desde una perspectiva de seguridad, este enfoque le permite auditar y controlar la forma y el momento en los que se usan estas cuentas. Evite la tentación de crear una sola cuenta de ejecución que proporcione permisos similares a los de administrador, ya que este enfoque no ofrece mucha protección contra el mal uso.

18.1.2 Activos y runbooks de Azure Automation

La cuenta de Azure Automation que creó en la sección 18.1.1 incluye algunos runbooks de ejemplo. Hay disponibles ejemplos de PowerShell y Python. Los activos y certificados de conexión también se agregan a la cuenta de Automation para las cuentas de ejecución que se crearon. Exploraremos esos activos de conexión compartida.

Pruébelo ahora

Para ver los activos configurados y los runbooks de ejemplo, complete los pasos siguientes:

- 1 En Azure Portal, seleccione Grupos de recursos a la izquierda; elija su grupo, como azuremolchapter18; y seleccione su cuenta de Azure Automation, como azuremol.
- 2 En Recursos compartidos que se encuentra en el menú a la izquierda, seleccione Conexiones.
- 3 Seleccione AzureRunAsConnection, como se muestra en la figura 18.2.

- 4 Seleccione Certificados en el menú principal de la cuenta de Automation en Recursos compartidos y, a continuación, elija AzureRunAsCertificate. Tal como se muestra en la figura 18.3, la huella digital coincide con RunAsConnection del paso anterior.

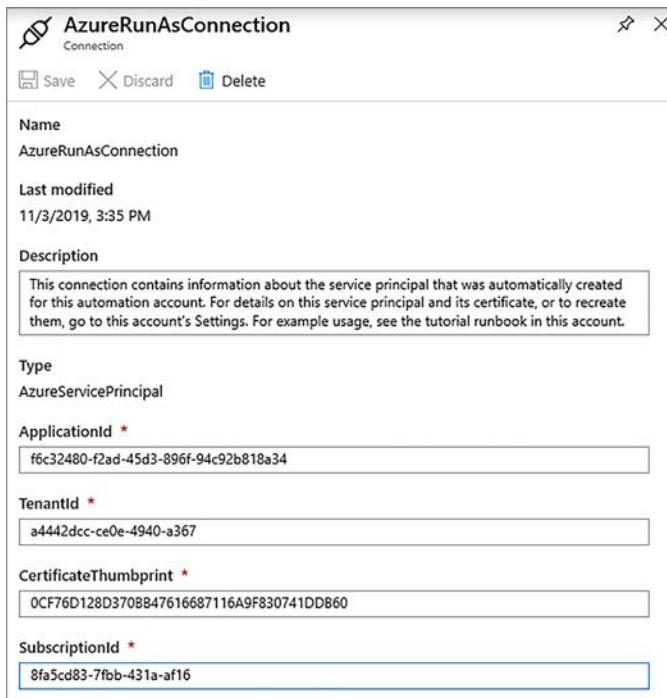


Figura 18.2 La información sobre la cuenta de ejecución incluye ApplicationId y TenantId, propiedades específicas para Azure AD que ayudan a identificar las credenciales de esta cuenta. Se muestra CertificateThumbprint, que coincide con un certificado digital que veremos en el paso siguiente.

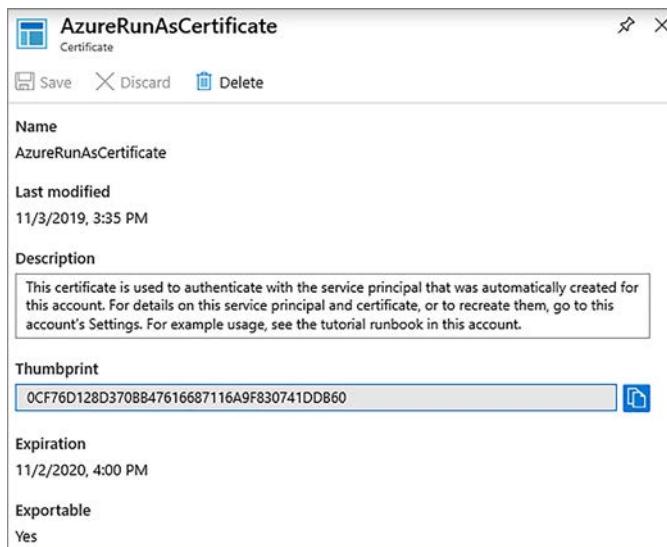


Figura 18.3 La huella digital de RunAsCertificate coincide con la que se muestra en RunAsConnection. En sus runbooks, usted define cuál activo de conexión usará. El certificado apropiado se usa para iniciar sesión en la cuenta Azure.

- 5 Ahora que entiende los activos para las conexiones y los certificados, observemos uno de los runbooks de ejemplo. Elija Runbooks en el menú que se encuentra a la izquierda en la cuenta de Automation. Hay disponibles algunos runbooks de ejemplo.
- 6 Elija el runbook de PowerShell llamado AzureAutomationTutorialScript.
- 7 En la parte superior del runbook de ejemplo hay opciones para iniciar, ver y editar el runbook. Estas opciones no requieren mayor explicación.

También tiene la opción de programar, que le permite crear o seleccionar un recurso compartido que define un programa para ejecutar el runbook en un momento determinado, y una opción para webhooks, que le permite crear una URL de webhook para ejecutar el runbook de algún otro script o acción. Elija Ver.

Azure Automation y control de origen con GitHub

Los runbooks se pueden integrar con un sistema de control de origen, como GitHub. Uno de los grandes beneficios de un sistema de control de origen para los runbooks es que proporciona una forma de documentar la administración de cambios y revertir a versiones anteriores de los runbooks en caso de un problema.

Cada vez que guarda un runbook de Azure Automation, se compromete una nueva versión para el control de origen. No es necesario que salga del editor de runbook, ya que la plataforma Azure y el sistema de control de origen configurado están configurados para avanzar y retroceder. Si tiene un problema con el nuevo runbook, puede extraer una versión anterior del control de origen que permita continuar con la ejecución de los trabajos sin retraso y luego solucionar el problema que tenga la versión actualizada.

El uso del control de origen también proporciona un registro de los cambios realizados y el momento en el que se hicieron. Si necesita auditar sus runbooks o entender cómo se desarrollaron con el tiempo, los sistemas de control de origen ofrecen una forma excelente de ver las diferencias con cada revisión.

18.2 Runbook de ejemplo de Azure Automation

Examinemos cómo el runbook de PowerShell de ejemplo, AzureAutomationTutorialScript, se conecta a Azure y reúne información sobre sus recursos. Puede seguir con el runbook de ejemplo de Python si lo prefiere; el diseño es similar. PowerShell y Python son los únicos lenguajes compatibles actualmente con los runbooks de Azure Automation. En la siguiente lista se configuran las credenciales de conexión en el runbook.

Listado 18.1 Configuración de credenciales de conexión

```
$connectionName = "AzureRunAsConnection" ← Se crea un objeto para $connectionName
try
{
    # Get the connection "AzureRunAsConnection"
    $servicePrincipalConnection=Get-AutomationConnection -Name ← Se hace la solicitud de conexión
    $connectionName                                ← Se crea un objeto de entidad de servicio
    "Logging in to Azure..."                         ` Add-AzureRmAccount
}
```

```

    -ServicePrincipal ` 
    -TenantId $servicePrincipalConnection.TenantId ` 
    -ApplicationId $servicePrincipalConnection.ApplicationId ` 
    -CertificateThumbprint
    ➔ $servicePrincipalConnection.CertificateThumbprint
}

```

Se inicia sesión en Azure

El código comienza con la creación de un objeto para \$connectionName. En el ejercicio “Pruébelo ahora”, vio que se creó un activo de conexión predeterminado para AzureRunAsConnection. A medida que cree sus propios runbooks, es posible que desee crear cuentas de ejecución adicionales y activos de conexión para separar los runbooks y las credenciales que usan. Las partes de conexión y el control de excepciones que veremos a continuación deben ser comunes en todos los runbooks. Según sea necesario, puede cambiar el activo de conexión de ejecución a usar.

A continuación, se usa una instrucción try para realizar la solicitud de conexión. Se crea un objeto de entidad de servicio llamado \$servicePrincipalConnection, basado en \$connectionName. Luego, el runbook inicia sesión en Azure con Add-AzureRmAccount y usa el objeto \$servicePrincipalConnection para obtener TenantId, ApplicationId y Certificate-Thumbprint. Analizamos estos parámetros anteriormente, como parte del activo de conexión. El activo de certificado que coincide con la huella digital de \$servicePrincipalConnection se usa para completar el inicio de sesión en Azure.

La siguiente lista muestra que, si la conexión falla, el runbook detecta el error y detiene la ejecución.

Listado 18.2 Detectar un error y detener la ejecución del runbook

```

catch {
    if (!$servicePrincipalConnection)
    {
        $ErrorMessage = "Connection $connectionName not found."
        throw $ErrorMessage
    } else{
        Write-Error -Message $_.Exception
        throw $_.Exception
    }
}

```

La instrucción catch manipula cualquier error como parte del intento de inicio de sesión. Si no se puede encontrar una conexión de entidad de servicio, se genera un error. Este error generalmente significa que no se puede encontrar el recurso de conexión que especificó. Compruebe el nombre y la ortografía de la conexión.

De lo contrario, se encontró el objeto de conexión y se utilizó la entidad de servicio para iniciar sesión, pero el proceso de autenticación no se realizó correctamente. Este error podría provenir de un certificado que ya no es válido o de una cuenta de ejecución que ya no está habilitada. Esta funcionalidad muestra cómo puede revocar una cuenta en Azure AD y garantizar que cualquier runbook que use las credenciales ya no pueda ejecutarse.

Ahora el runbook obtiene una lista de todos los recursos de Azure.

Listado 18.3 Obtener una lista de los recursos de Azure

```
$ResourceGroups = Get-AzureRmResourceGroup
foreach ($ResourceGroup in $ResourceGroups)
{
    Write-Output ("Showing resources in resource group "
    + $ResourceGroup.ResourceGroupName)
    $Resources = Find-AzureRmResource -ResourceGroupNameContains
    $ResourceGroup.ResourceGroupName |
    Select ResourceName, ResourceType
    ForEach ($Resource in $Resources)
    {
        Write-Output ($Resource.ResourceName + " of type "
        + $Resource.ResourceType)
    }
    Write-Output ("")
}
```

La parte final del runbook es donde iría su código de runbook. Se crea un objeto para \$ResourceGroups que obtiene una lista de todos los grupos de recursos de Azure disponibles. Luego, un bucle foreach recorre los grupos de recursos, encuentra una lista de recursos y escribe una lista de los nombres y tipos de recursos.

En este ejemplo básico se muestra cómo puede interactuar con Azure cuando el runbook se ha autenticado con la suscripción. Si implementa RBAC en la cuenta de ejecución, solo se devuelven los grupos de recursos que la cuenta tiene permisos para ver. Este enfoque de RBAC destaca por qué es un buen principio de seguridad crear y usar cuentas de ejecución con un alcance para limitar el acceso de los runbooks a los recursos en su entorno de Azure. Siempre trate de proporcionar la menor cantidad de privilegios necesarios.

Si todo esto de PowerShell o Python es nuevo para usted, no se preocupe. Ambos proporcionan un excelente lenguaje de script básico, pero también pueden usarse para desarrollar aplicaciones complejas y potentes. Como desarrollador, cualquiera de los dos lenguajes debiera ser relativamente fácil de aprender y usar. Si usted es un profesional de TI, las tareas de automatización le ayudan a tener tiempo libre para realizar todos los demás trabajos acumulados, y tanto PowerShell como Python son buenos lugares para comenzar. Manning Publications también posee otros grandes libros que le serán de ayuda.

18.2.1 Ejecución y visualización de la salida de un runbook de ejemplo

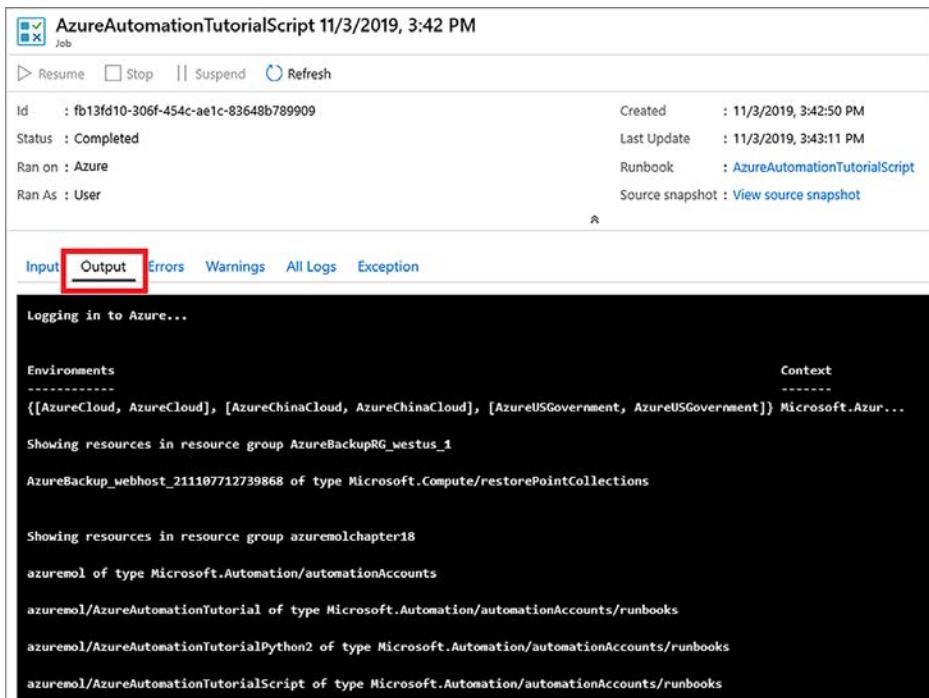
Ahora que ha visto lo que contiene el script de runbook de ejemplo y cómo se usan los activos de conexión y certificado, ejecutemos el runbook y observemos el resultado.

Pruébelo ahora

Para ver el runbook en acción, complete los pasos siguientes:

- 1 Cierre la ventana que muestra el contenido del runbook y vuelva a la descripción general de AzureAutomationScriptTutorial.

- 2 Seleccione Iniciar en la parte superior de la ventana del runbook.
- 3 Confirme que desea iniciar el runbook y espere unos segundos para que comience a ejecutarse.
- 4 Seleccione Salida, tal como se muestra en la figura 18.4, y luego observe la ventana de la consola a medida que el runbook inicia sesión en Azure, obtiene una lista de grupos de recursos, pasa reiteradamente y muestra la lista de recursos en cada uno.



The screenshot shows the Azure Automation Runbook interface. At the top, there's a header bar with a checkmark icon, the text 'AzureAutomationTutorialScript 11/3/2019, 3:42 PM', and a 'Job' button. Below the header are buttons for 'Resume', 'Stop', 'Suspend', and 'Refresh'. The main area displays runbook details: Id (fb13fd10-306f-454c-ae1c-83648b789909), Status (Completed), Ran on (Azure), Ran As (User), Created (11/3/2019, 3:42:50 PM), Last Update (11/3/2019, 3:43:11 PM), Runbook (AzureAutomationTutorialScript), and Source snapshot (View source snapshot). A red box highlights the 'Output' tab in the navigation bar below the details. The 'Output' tab is selected, showing a log window with the following content:

```

Logging in to Azure...

Environments
-----
{{[AzureCloud, AzureCloud], [AzureChinaCloud, AzureChinaCloud], [AzureUSGovernment, AzureUSGovernment]}} Microsoft.Azure...
Context
-----
Showing resources in resource group AzureBackupRG_westus_1
AzureBackup_webhost_211107712739868 of type Microsoft.Compute/restorePointCollections

Showing resources in resource group azuremolchapter18
azuremol of type Microsoft.Automation/automationAccounts
azuremol/AzureAutomationTutorial of type Microsoft.Automation/automationAccounts/runbooks
azuremol/AzureAutomationTutorialPython2 of type Microsoft.Automation/automationAccounts/runbooks
azuremol/AzureAutomationTutorialScript of type Microsoft.Automation/automationAccounts/runbooks

```

Figura 18.4 Puede ver la salida del runbook, junto con cualquier registro que se genere o con errores y advertencias. Este ejemplo básico se completa en unos pocos segundos, pero los runbooks más complejos pueden tardar más. Puede monitorear el estado de esos runbooks más largos y detener o pausar su ejecución según sea necesario.

No es necesario que los runbooks de automatización existan de forma aislada. Un runbook puede ejecutar otro runbook. Esta capacidad le permite crear una automatización compleja y de varios pasos, y minimizar la duplicación de código. A medida que diseña y crea runbooks, intente dividirlos en pequeños bloques de código discretos. Las funciones comunes que puede reutilizar, como iniciar sesión en Azure y generar una lista de recursos o una lista de VM, deben crearse como pequeños runbooks que se pueden incluir en los runbooks más grandes. Cuando se lanzan nuevos cmdlets de PowerShell o se cambian los parámetros, puede actualizar rápidamente un solo runbook compartido que incluya esos cmdlets en lugar de tener que actualizar múltiples runbooks diferentes. Al principio, puede parecer que no vale la pena hacer un poco de trabajo extra con runbooks más pequeños y reutilizables, pero a medida que su

entorno y el uso de la Automatización crezcan, me lo agradecerá. Gran parte de lo que ha hecho en este libro ha sido en implementaciones más pequeñas, pero comience a pensar en cómo implementar y administrar aplicaciones a escala.

18.3 Desired State Configuration (DSC) de PowerShell

En el capítulo 12 se introdujo el concepto de extensiones de VM. Una *extensión* es un pequeño componente de software que se instala en una VM para realizar una tarea determinada. La extensión de diagnóstico de VM se instaló en una VM para permitir que las métricas de rendimiento y los registros de diagnóstico se informen a la plataforma Azure desde la VM. Eso es genial, pero también hablamos un poco sobre cómo puede instalar software automáticamente.

Una forma de instalar software y configurar un servidor es usar la Desired State Configuration (DSC) de PowerShell. Con DSC, usted define cómo desea que se configure un servidor: el estado deseado. Puede definir los paquetes que se instalarán, las funciones que se configurarán o los archivos que se crearán, por ejemplo. Lo bueno de la DSC es que va más allá de la primera acción de instalación y configuración. Con el tiempo, los servidores a menudo se someten a eventos de mantenimiento o solución de problemas donde las configuraciones y paquetes se cambian manualmente. Luego, el servidor se desviará del estado deseado que usted definió inicialmente. En la figura 18.5 se muestra cómo Azure Automation puede actuar como un servidor central que almacena las definiciones de DSC, permitiendo a los servidores de destino recibir sus configuraciones e informar sobre su cumplimiento.

El administrador de configuración local (LCM) en cada servidor de destino controla el proceso para conectarse al servidor de extracción de Azure Automation, recibir y analizar la definición de DSC, y aplicar e informar sobre el cumplimiento. El motor LCM puede funcionar sin un servidor de extracción, se llama localmente al proceso

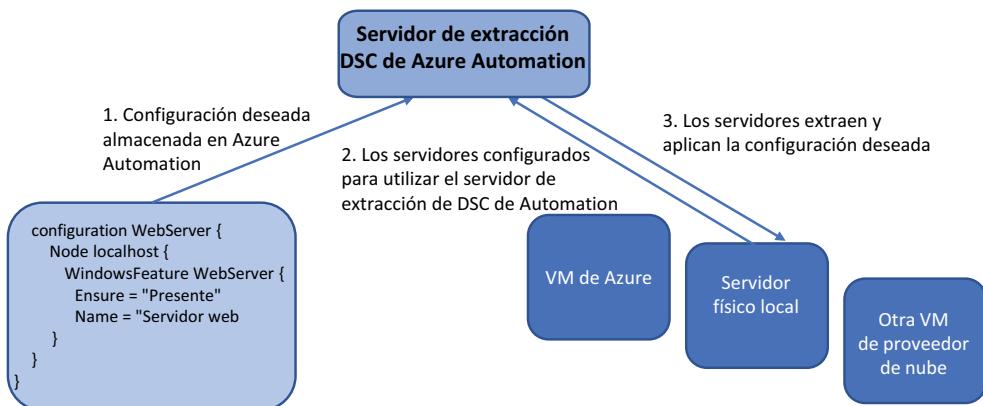


Figura 18.5 La configuración de estado deseada para un servidor se crea y almacena en Azure Automation. La cuenta de Automation actúa como un servidor de extracción, que permite a los servidores conectados extraer la configuración requerida desde una ubicación central. Se pueden establecer diferentes modos de configuración para el comportamiento de corrección del servidor si su configuración se desvía del estado deseado.

para leer y aplicar una definición de DSC. En este modo, donde usted configura manualmente el motor LCM, pierde muchos de los controles e informes centrales que a menudo se necesitan cuando administra muchos servidores.

También hay flexibilidad en la forma en la que los servidores de destino procesan las definiciones de DSC recibidas del servidor de extracción de Azure Automation. Puede configurar la DSC para que funcione en uno de los tres modos de configuración:

- *Solo aplicar*: su estado deseado se envía y se aplica al servidor de destino, eso es todo. Esto es como el comportamiento de Azure Custom Script Extension en el sentido de que cualquier configuración o instalación se aplica cuando se implementa por primera vez, pero no hay procesos establecidos para detener esas configuraciones que cambian manualmente durante el ciclo de vida del servidor.
- *Aplicar y monitorear*: después de que el servidor tiene aplicado el estado deseado, la DSC continúa monitoreando cualquier cambio que haga que el servidor se desvíe de esa configuración inicial. Se puede usar un informe central para ver los servidores que ya no cumplen con el estado deseado. Esta configuración es un buen acuerdo entre la necesidad de mantener un servidor que cumpla con el estado deseado y proporcionar un elemento de interacción humana para tomar decisiones relacionadas con las opciones de corrección.
- *Aplicar y autocorregir*: se aplica la configuración más automatizada y autónoma al estado deseado y luego se supervisa cualquier desviación y se corrige automáticamente el servidor en caso de que se produzcan cambios para garantizar que sigue siendo compatible. Existe el peligro de que los cambios manuales legítimos se sobrescriban y, en cambio, se vuelva al estado deseado configurado, pero este modo de configuración garantiza que las configuraciones que asigne siempre tengan prioridad.

La DSC de PowerShell se puede usar en VM que se ejecutan en otros proveedores de nube, así como en VM locales y servidores virtuales. Gracias a .NET Core, la DSC de PowerShell también se puede usar en servidores Linux, por lo que no es una solución exclusiva para Windows. Esta compatibilidad con múltiples SO y proveedores hace de PowerShell una opción poderosa para configurar y administrar servidores a escala.

Puede crear y mantener su propio servidor de extracción de DSC, pero las características integradas de Azure Automation ofrecen algunos beneficios adicionales:

- Las credenciales se administran de forma centralizada y los certificados se generan automáticamente.
- La comunicación entre el servidor de extracción de DSC y los servidores de destino está cifrada.
- Se proporcionan informes incorporados para el cumplimiento de DSC, y existe una integración con Log Analytics para generar informes y alertas más detallados.

Esta sección es un curso intensivo en DSC de PowerShell, un componente que es poderoso por sí mismo y ha estado ampliamente disponible desde hace algunos años. Cuando se combina con Azure Automation, la DSC es una excelente opción para automatizar la instalación y configuración del software. Piense en los capítulos anteriores sobre conjuntos de escala de máquinas virtuales, por ejemplo. Puede aplicar una configuración de DSC al conjunto de escalas con Azure Automation, y luego, a medida que se crea cada VM en el conjunto de escalas, se configurará automáticamente con los componentes y archivos de aplicación necesarios.

18.3.1 Definición y uso de DSC de PowerShell y un servidor de extracción de Azure Automation

¡Espero que este arrollador recorrido por la DSC de PowerShell le haya dado una idea de lo que es posible! Vamos a usar la DSC de PowerShell para automatizar el ejemplo de instalación de un servidor web básico en una VM.

Pruébelo ahora

Complete los siguientes pasos para ver PowerShell DSC en acción:

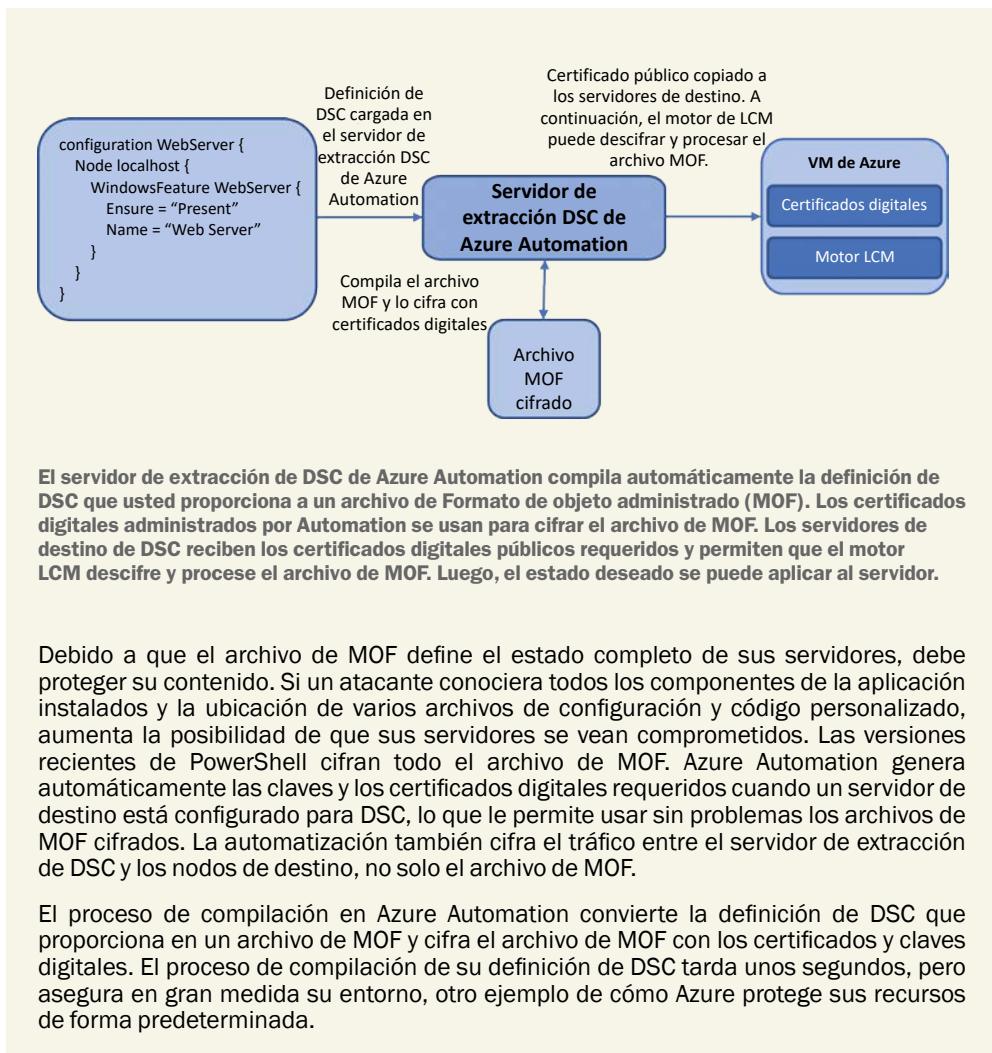
- 1 Cree una VM de Windows Server 2019 Datacenter y abra el puerto TCP 80 para el tráfico HTTP. ¡Puede hacerlo solo, estamos en el capítulo 18! Puede crear la VM en Cloud Shell o en Azure Portal, usted decide. Use el grupo de recursos que creó en los ejercicios anteriores, como `azuremolchapter18`. Puede continuar con los siguientes pasos a medida que se implementa la VM.
- 2 En su equipo local, cree un archivo llamado `webserver.ps1`; escriba el siguiente código; y guarde y cierre el archivo cuando haya terminado:

```
configuration WebServer {
    Node localhost {
        WindowsFeature WebServer {
            Ensure = "Present"
            Name = "Web-Server"
        }
    }
}
```

- 3 En Azure Portal, seleccione su grupo de recursos y luego elija su cuenta de Automation.
- 4 A la izquierda, elija State Configuration (DSC); Seleccione la pestaña Configuraciones; y en la parte superior de la ventana, elija Agregar una configuración.
- 5 Busque y seleccione su archivo `webserver.ps1`. El nombre de la configuración debe coincidir con el nombre del archivo, así que acepte el nombre predeterminado del servidor web y luego elija Aceptar.
La configuración tardará unos momentos en cargarse y crearse.
- 6 Cuando esté listo, seleccione la configuración de la lista y luego elija Compilar.

La DSC en segundo plano

Hagamos una pausa para hablar sobre lo que sucede cuando compila la configuración, tal como se muestra en la figura en esta barra lateral. Para distribuir las definiciones de DSC, sus archivos de PowerShell se convierten en un archivo de Formato de objeto administrado (MOF). Este tipo de archivo se usa para más que solo DSC de PowerShell y permite cambios de configuración en los componentes de Windows de una manera central y bien conocida. Cualquier definición de DSC, no solo en Azure Automation, debe compilarse antes de que pueda aplicarse a un servidor de destino. El motor LCM solo acepta y procesa archivos de MOF.



- 7 Para aplicar la configuración a su VM, seleccione la pestaña de nodos en las ventanas State Configuration (DSC); seleccione Agregar; y elija la VM que creó en los pasos anteriores.
- 8 Elija Conectar.
- 9 En el menú desplegable Nombre de configuración del nodo, elija webserver.localhost.
- 10 Establezca el modo de configuración en ApplyAndMonitor y seleccione Aceptar.
Puede demorar uno o dos minutos en habilitar la VM para usar el servidor de extracción de DSC de Azure PowerShell y aplicar el estado inicial deseado.
- 11 Cuando Azure Portal informa que la configuración está aplicada, seleccione su grupo de recursos; a continuación, seleccione la VM que creó en los pasos anteriores.

- 12 ¿Abrió el puerto TCP 80 para la VM cuando la creó? Si no es así, cree una regla de grupo de seguridad de red para permitir el tráfico y luego abra la IP pública de la VM en un navegador web. El proceso de DSC instala el servidor web IIS y la página web predeterminada se carga, tal como se muestra en la figura 18.6.

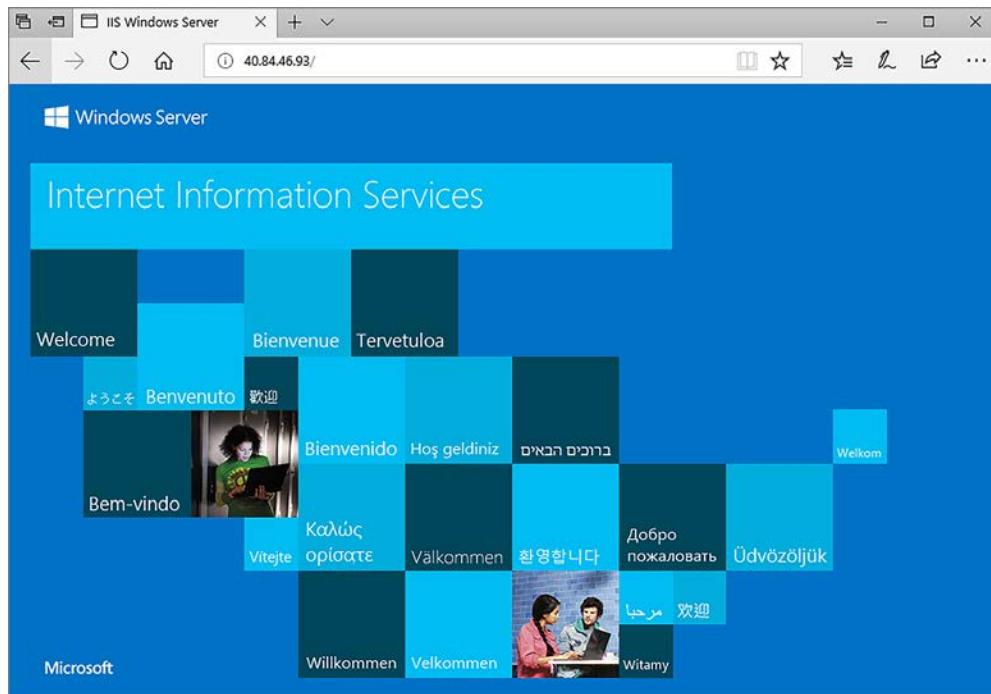


Figura 18.6 Despues de que la VM se ha conectado a la DSC de Azure Automation, se aplica el estado deseado y se instala el servidor web de IIS.

Este ejemplo básico de DSC de PowerShell solo instala la característica de servidor web. Puede usar la DSC de PowerShell para configurar el servidor web de IIS o copiar el código de su aplicación a la VM y ejecutar el sitio. Se pueden usar definiciones de DSC complejas para preparar la VM para servir el tráfico a los clientes de su pizzería sin interacción manual. Nuevamente, recuerde cómo debe diseñar sus aplicaciones para que se amplíen automáticamente: ¡la VM no puede esperar a que alguien inicie sesión, instale y configure todo manualmente!

18.4 Laboratorio: Uso de DSC con Linux

Solo para demostrar que DSC de PowerShell funciona en servidores Linux, creamos una VM de Ubuntu, instalamos los requisitos previos necesarios y luego instalamos un servidor web básico de NGINX con DSC. En producción, podría usar una imagen de VM personalizada que ya tuviera instalados los componentes de administración y luego aplicar las definiciones de DSC de PowerShell de la forma habitual:

- 1 DSC de PowerShell para Linux tiene algunas limitaciones en las distribuciones de Linux que admite sin configuración adicional, así que para mantener este ejercicio de laboratorio de fin de capítulo tan simple como sea posible, cree una VM CentOS 7.7 o posterior, y abra el puerto 80.
- 2 En la cuenta de Azure Automation, elija Módulos en el menú que se encuentra a la izquierda.
- 3 Seleccione Explorar galería y luego busque, seleccione e importe el módulo nx para administrar los recursos de DSC de Linux.
- 4 En su equipo local, cree un archivo llamado httpd.ps1 y escriba el siguiente código:

```
configuration httpd {
    Import-DSCResource -Module nx
    Node localhost {
        nxPackage httpd {
            Name = "httpd"
            Ensure = "Present"
            PackageManager = "yum"
        }
        nxService httpd {
            Name = "httpd"
            State = "running"
            Enabled = $true
            Controller = "systemd"
        }
    }
}
```

- 5 Agregue una configuración DSC a la cuenta de Azure Automation, suba el archivo httpd.ps1 fil y compile la configuración.
- 6 Agregue un nodo de DSC a su cuenta de Azure Automation, seleccione su VM CentOS, y luego elija su nombre de configuración de nodo httpd.localhost.

Nuevamente, la VM tarda uno o dos minutos en aplicar la configuración deseada. Puede ver la lista de VM conectadas y su estado de cumplimiento en la ventana de nodos de DSC. La VM informa de que cumple cuando el LCM ha aceptado y aplicado el archivo MOF, pero los comandos para instalar y configurar los paquetes httpd necesarios dentro de la VM pueden tardar uno o dos minutos más.

- 7 Seleccione su VM CentOS en Azure Portal, obtenga su dirección IP pública e introduzca la dirección IP de su VM en un navegador web para ver el servidor web instalado por DSC. Si el sitio web no se carga, espere uno o dos minutos para que finalice el proceso de instalación y luego actualice la página.

Si desea experimentar verdaderamente el nuevo mundo de Microsoft y Linux, puede instalar PowerShell en su VM de Linux. Complete los pasos de configuración rápida en <http://mng.bz/VgyP> para entender cómo pueden ser ahora los scripts multiplataforma de PowerShell.

Contenedores de Azure

Los contenedores, Docker y Kubernetes han ganado una enorme cantidad de seguidores en algunos años. De la misma manera que la virtualización de servidores comenzó a cambiar la forma en que los departamentos de TI dirigían sus centros de datos a mediados de la década de 2000, las herramientas de contenedores modernas y los orquestadores ahora están sacudiendo la forma en que desarrollamos y ejecutamos aplicaciones. No hay nada que conecte inherentemente el crecimiento de los contenedores con la informática en la nube, pero cuando se combinan, proporcionan una gran manera de desarrollar aplicaciones con un enfoque nativo de nube. Se han escrito libros enteros en Docker y Kubernetes, pero continuemos con una introducción relámpago para ver cómo se pueden ejecutar con rapidez los contenedores en Azure. Hay un conjunto poderoso de servicios Azure dedicados a los contenedores que se alinea más con el enfoque PaaS. Puede centrarse en la forma de crear y ejecutar las aplicaciones, en lugar de cómo administrar la infraestructura de contenedores, la orquestación y los componentes de clúster.

En este capítulo, examinamos qué son los contenedores, cómo se involucró Dockery qué puede hacer Kubernetes por usted. Para ver cómo ejecutar rápidamente una instancia de contenedor único o varias instancias de contenedor en un clúster, exploramos Azure Container Instances (ACI) y Azure Kubernetes Service (AKS).

19.1 ¿Qué son los contenedores?

En los últimos años, ha habido una gran ola de interés y adopción en torno a los contenedores y me impresionaría si usted no ha oído hablar de una empresa que ha liderado esta carga: Docker. Pero, ¿qué es exactamente un contenedor, y qué tiene que ver Docker con este?

En primer lugar, analicemos un host de virtualización tradicional que ejecuta VM. La figura 19.1 es como el diagrama que miramos antes en el capítulo 1, donde cada máquina virtual tiene su propio hardware virtual y sistema operativo invitado.

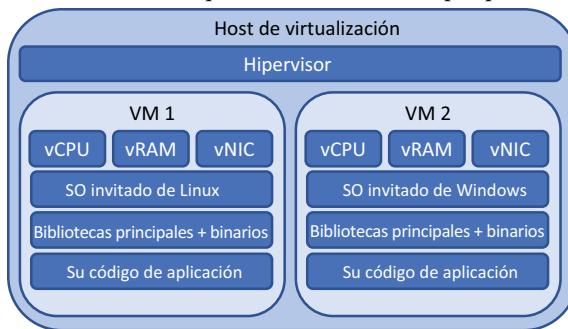


Figura 19.1 Con una infraestructura tradicional de VM, el hipervisor de cada host de virtualización proporciona una capa de aislamiento al entregar a cada máquina virtual su propio conjunto de dispositivos de hardware virtuales, como una CPU virtual, RAM virtual y NIC virtuales. La VM instala un SO invitado, como Ubuntu, Linux o Windows Server, que puede utilizar este hardware virtual. Finalmente, instala su aplicación y cualquier biblioteca necesaria. Este nivel de aislamiento hace que las máquinas virtuales sean muy seguras, pero agrega una capa de sobrecarga en términos de recursos de procesamiento, almacenamiento y tiempos de inicio.

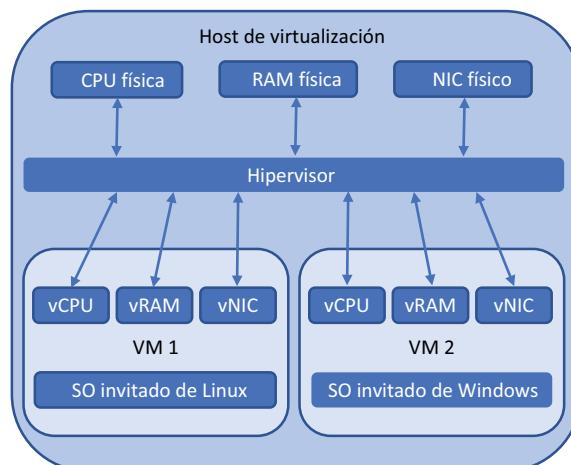
Un contenedor elimina el hardware virtual y el sistema operativo invitado. Todo lo que se incluye en un contenedor son las principales aplicaciones y bibliotecas necesarias para ejecutar la aplicación, como se muestra en la figura 19.2.

Muchas VM pueden ejecutarse en un solo hipervisor, cada máquina virtual con su propio sistema operativo virtual invitado, hardware virtual y pila de aplicaciones. El hipervisor administra las solicitudes del hardware virtual de cada máquina virtual, programa la asignación y el uso compartido de esos recursos de hardware físico, y aplica la seguridad y el aislamiento de cada máquina virtual. El trabajo del hipervisor se muestra en la figura 19.3.

Figura 19.3 En un host de VM tradicional, el hipervisor proporciona la programación de las solicitudes desde el hardware virtual en cada máquina virtual del hardware y la infraestructura física subyacente. Normalmente, el hipervisor no tiene conciencia de las instrucciones específicas que el sistema operativo invitado está planificando en el tiempo físico de la CPU, solo que se requiere tiempo de CPU.



Figura 19.2 Un contenedor contiene solo las bibliotecas básicas, los binarios y el código de aplicación necesarios para ejecutar una aplicación. El contenedor es ligero y portátil, ya que elimina el sistema operativo invitado y la capa de hardware virtual, lo que también reduce el tamaño en disco del contenedor y los tiempos de inicio.



También se pueden ejecutar varios contenedores en un único host. El host de contenedor recibe las distintas llamadas de sistema de cada contenedor y programa la asignación y la distribución de esas solicitudes en un kernel de base compartida, sistema operativo y recursos de hardware. Los contenedores proporcionan un aislamiento lógico de los procesos de aplicación. El trabajo del tiempo de ejecución del contenedor se muestra en la figura 19.4.

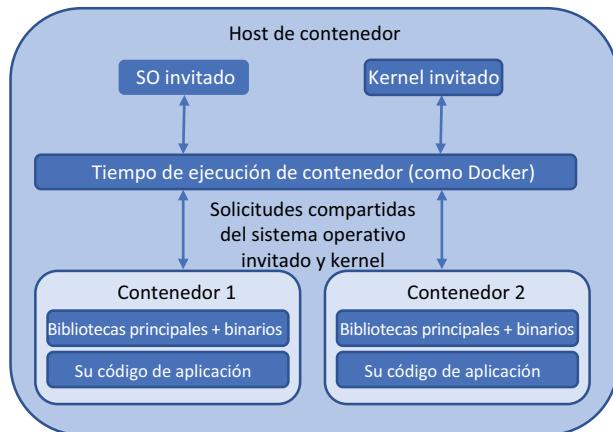


Figura 19.4 Los contenedores
tienen un sistema operativo y kernel común. El tiempo de ejecución de contenedor controla las solicitudes de los contenedores al kernel compartido. Cada contenedor se ejecuta en un espacio de usuario aislado, y algunas características adicionales de seguridad protegen los contenedores entre sí.

Los contenedores suelen ser mucho más livianos que las máquinas virtuales. Los contenedores pueden iniciarse más rápido que las máquinas virtuales, a menudo en cuestión de segundos en lugar de minutos. El tamaño de una imagen de contenedor típicamente es de solo decenas o cientos de MB, en comparación con las muchas decenas de GB para VM. Todavía hay límites de seguridad y controles vigentes, pero es importante recordar que cada contenedor comparte técnicamente el kernel que otros contenedores en el mismo host.

Pruébelo ahora

Se tarda unos minutos en crear un clúster de AKS para su uso en los próximos ejercicios, así que complete los siguientes pasos y continúe leyendo el capítulo:

- 1 Abra Azure Portal y seleccione el ícono de Cloud Shell en el menú superior.
- 2 Cree un grupo de recursos. Proporcione un nombre, como azuremolchapter19, y una ubicación, como eastus. La disponibilidad de la región de AKS puede variar, así que seleccione una región principal como eastus o westeurope. (Para obtener una lista actualizada de la disponibilidad de la región, consulte <https://azure.microsoft.com/regions/services>).

```
az group create --name azuremolchapter19 --location eastus
```

- 3 Para crear un clúster Kubernetes, especifique --node-count como 2 y utilice conjuntos de escalado y zonas de disponibilidad (que aprendió en los capítulos anteriores):

```
az aks create \
    --resource-group azuremolchapter19 \
    --name azuremol \
    --node-count 2 \
    --vm-set-type VirtualMachineScaleSets \
    --zones 1 2 3 \
    --no-wait
```

El parámetro final `--no-wait` devuelve el control a Cloud Shell mientras se crea el resto del clúster. Siga leyendo mientras se implementa el clúster.

Docker se unió a la parte contenedora con un conjunto de herramientas y formatos estándar que definían cómo construir y ejecutar un contenedor. Docker se basa en las principales funcionalidades existentes de nivel de kernel de Linux y Windows para proporcionar una experiencia de contenedor sistemática y portátil entre plataformas. Un desarrollador puede crear un contenedor Docker en su equipo portátil que ejecuta macOS, validar y probar su aplicación y, a continuación, ejecutar exactamente el contenedor de Docker, sin modificación, en un clúster Linux o Windows más tradicional de manera local o en Azure. Todos los binarios de aplicación, bibliotecas y archivos de configuración requeridos se empaquetan como parte del contenedor, por lo que el sistema operativo host subyacente no se convierte en un factor de diseño o restricción.

La importancia de Docker no se debe perder aquí. Los términos *contenedores Docker* se utilizan a menudo indistintamente, aunque eso no es técnicamente exacto. Docker es un conjunto de herramientas que ayuda a los desarrolladores a compilar y ejecutar contenedores de forma sistemática, confiable y portátil. La facilidad de uso de estas herramientas llevó a la adopción rápida y cambió la forma que tenía la tecnología de contenedor subyacente por más de una década en la corriente principal. Los desarrolladores adoptaron los contenedores y la plataforma Docker, y los departamentos de TI han tenido que jugar a ponerse al día desde entonces.

Docker participa en la Open Container Initiative. El formato y las especificaciones que Docker definió para la forma de empaquetar y ejecutar un contenedor fueron algunos de los principios fundadores de este proyecto. El trabajo de Docker ha continuado y se basa en el trabajo realizado por otros. Los grandes colaboradores en el espacio de contenedores incluyen a IBM y Red Hat, aportando algunos de los diseños principales y el código que impulsa las plataformas de contenedores actuales. Es importante Open Container Initiative y el formato de diseño para el empaquetado de contenedores y tiempos de ejecución porque cada proveedor pone en capas sus propias herramientas en la parte superior de los formatos comunes, lo que les permite mover el contenedor subyacente entre las plataformas y tener la misma experiencia básica.

19.2 El enfoque de los microservicios a las aplicaciones

Si los contenedores ofrecen un concepto de aislamiento similar a las máquinas virtuales, ¿puede ejecutar el mismo tipo de cargas de trabajo que realiza en una VM? Bueno, sí y no. Solo porque puede hacer algo no significa necesariamente que debe hacerlo. Los contenedores se pueden utilizar para ejecutar cualquier carga de trabajo con la que se sienta cómodo, y hay ventajas en términos de portabilidad y funciones de orquestación que examinamos más adelante en la sección 19.4. Para maximizar los beneficios de los contenedores y prepararse para el éxito, aproveche la oportunidad de adoptar un modelo mental ligeramente diferente cuando comience a trabajar con contenedores. La figura 19.5 compara el modelo de aplicación tradicional con un enfoque de microservicios.

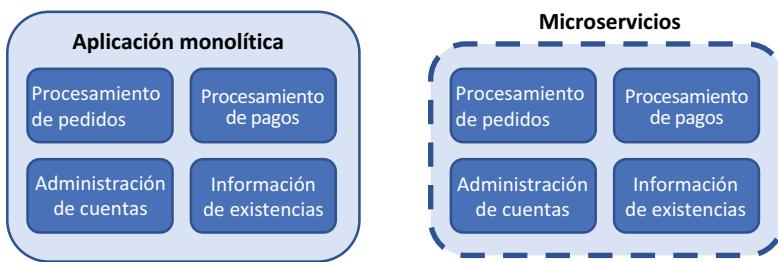


Figura 19.5 En una aplicación monolítica tradicional, toda la aplicación se ejecuta como una sola aplicación. La aplicación puede tener varios componentes, pero se ejecuta desde una sola instalación y se corrige y actualiza como una sola instancia. Con los microservicios, cada componente se desglosa en su propio servicio de aplicación y unidad de ejecución. Cada componente puede actualizarse, corregirse y escalarse independientemente de los demás.

Una VM estándar incluye una instalación completa del sistema operativo invitado, como Ubuntu o Windows Server. Esta instalación de sistema operativo base incluye cientos de componentes, bibliotecas y herramientas. A continuación, instala más bibliotecas y aplicaciones, como el servidor web NGINX o Microsoft SQL Server. Por último, se implementa el código de la aplicación. Esta VM normalmente ejecuta una parte grande, si no toda, de la aplicación. Es una instancia grande de instalación y ejecución de aplicación. Para mejorar el rendimiento, puede agregar más memoria o CPU a la VM (escala vertical, analizada en capítulos anteriores) o aumentar el número de instancias que ejecutan la aplicación (escala horizontal, como con conjuntos de escala). La creación de varias instancias de aplicación solo funciona si la aplicación es compatible con clústeres y, a menudo, implica algún tipo de almacenamiento de información compartido para permitir un estado uniforme en todas las instancias de la aplicación. Esta forma tradicional de implementación se denomina aplicación *monolítica*.

Un enfoque diferente de cómo diseñar, desarrollar y ejecutar aplicaciones es dividir las cosas en componentes más pequeños y de tamaño de bite. Se trata de un enfoque de *microservicios* para el desarrollo y la implementación de aplicaciones. Cada microservicio es responsable de una pequeña parte del entorno de aplicación más amplio. Los microservicios pueden crecer, escalar y actualizarse independientemente del resto del entorno de aplicación.

Aunque este modelo puede ofrecer desafíos al principio mientras que los equipos de desarrollo y TI aprenden a adoptar una manera diferente de construir e implementar aplicaciones, los contenedores son un gran ajuste para el enfoque de microservicios. Los desarrolladores tienen el poder de implementar actualizaciones más pequeñas y más incrementales a un ritmo más rápido que el enfoque monolítico del desarrollo de aplicaciones. Los microservicios y los contenedores también son una excelente opción para los flujos de trabajo de integración continua y entrega continua (CI/CD) que facilitan que desarrolle, pruebe, escale e implemente las actualizaciones. Sus clientes reciben nuevas funcionalidades o correcciones de errores con más rapidez de lo que lo harían e, idealmente, es de esperar que su negocio crezca como resultado.

Microservicios con Azure Service Fabric

Este capítulo se centra principalmente en los contenedores Docker y la orquestación con Kubernetes, pero un servicio Azure similar traslada el desarrollo de aplicaciones hacia un modelo de microservicios. Azure Service Fabric ha existido durante varios años y fue

históricamente un enfoque centrado en Windows para el desarrollo de aplicaciones donde cada componente se desglosa en su propio microservicio. Service Fabric mantiene un registro de dónde se ejecuta cada componente de microservicio en un clúster, permite que los servicios se descubran y se comuniquen entre sí, y que manipulen la redundancia y el escalamiento.

Muchos servicios de Azure grandes utilizan Service Fabric "por debajo", incluido Cosmos DB. Eso debe darle una sensación de la capacidad y poder que puede tener Service Fabric. Service Fabric en sí se ejecuta en la parte superior de los conjuntos de escala de máquinas virtuales. Sabe una o dos cosas sobre los conjuntos de escalas por ahora, ¿verdad?

La plataforma Service Fabric ha madurado, y ahora puede manipular tanto Windows como Linux como el sistema operativo invitado, para que pueda desarrollar su aplicación con cualquier lenguaje de programación con el que se sienta cómodo. Este es otro ejemplo de elección en Azure: tiene la flexibilidad de elegir cómo quiere administrar y orquestar sus aplicaciones de contenedor. Tanto Service Fabric como AKS tienen excelentes beneficios y casos de uso.

Como buen punto de partida, si actualmente desarrolla, o desea desarrollar, los microservicios fuera de los contenedores, Service Fabric es una gran opción. Las aplicaciones diseñadas en torno al modelo de actor también encajan perfectamente, ya que Service Fabric se creó originalmente con este modelo de programación en mente. Service Fabric proporciona un enfoque unificado para manipular aplicaciones de microservicios más tradicionales y aplicaciones basadas en contenedores. Si decide adoptar contenedores para otras cargas de trabajo, puede utilizar las mismas herramientas de administración e interfaz de Service Fabric para administrar todos los entornos de aplicaciones.

Para un enfoque de aplicación más centrado en contenedores desde el primer momento, AKS puede ser una mejor opción, con el crecimiento y la adopción de Kubernetes que proporciona una experiencia de contenedores de primera clase. Puede ejecutar contenedores de Linux y Windows en AKS.

19.3 Azure Container Instances

Ahora que entiende un poco más acerca de lo que son los contenedores y cómo se pueden utilizar, profundizaremos y crearemos una instancia básica de la pizzería. Este ejemplo es el mismo usado en capítulos anteriores, donde creó una VM básica que ejecutó su sitio web, o implementó la aplicación para aplicaciones web. En ambos casos, había que crear la VM o la aplicación web, conectarse a ella y, a continuación, implementar una página web básica. ¿Puede el poder de los contenedores hacer que su vida sea mucho más fácil? Desde luego.

Un servicio definido llamado el primer momento (ACI) le permite crear y ejecutar contenedores en cuestión de segundos. No hay recursos de red por adelantado para crear y configurar, y se paga por cada instancia de contenedor por la segunda. Si nunca ha usado contenedores y no quiere instalar nada localmente en el equipo, ACI es una gran manera de probar la tecnología.

Para ver cómo puede ejecutar rápidamente su pizzería, vamos a crear una instancia de contenedor. Solo se necesita un comando para ejecutar una instancia de contenedor, pero la figura 19.6 muestra cómo reunir muchos componentes para hacer que esto tras-

bambalinas. Observaremos los componentes de Dockerfile y Docker Hub después de que la instancia de contenedor esté en funcionamiento.

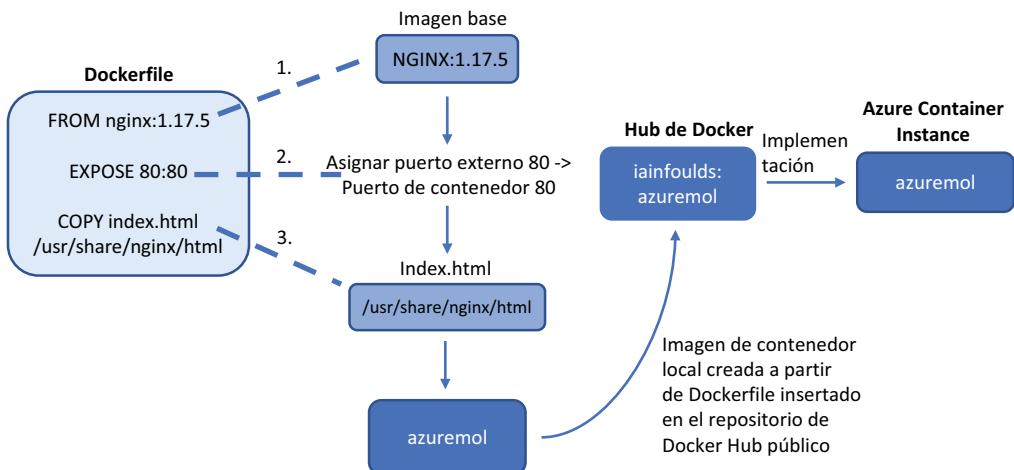


Figura 19.6 Dockerfile se utilizó para desarrollar una imagen completa del contenedor: azuremol. Esta imagen se llevó a un registro público en línea llamado Docker Hub. Ahora puede crear una instancia de contenedor mediante esta imagen pública precompilada desde Docker Hub, que proporciona una imagen de aplicación lista para ejecutarse.

Pruébelo ahora

Para crear una instancia de contenedor de Azure que ejecute un sitio web básico, complete los pasos siguientes.

- 1 Abra Azure Portal y seleccione el icono de Cloud Shell en el menú superior.
- 2 Cree una instancia de contenedor y especifique que desea tener una dirección IP pública y abra el puerto 80:

```
az container create \
--resource-group azuremolchapter19 \
--name azuremol \
--image iainfoulds/azuremol \
--ip-address public \
--ports 80
```

En este ejercicio se usa una imagen de ejemplo que creé para usted, la que examinaremos un poco más cuando el contenedor esté en funcionamiento.

- 3 Para ver lo que se creó, observe la salida del comando para crear el contenedor.

En la sección Eventos, puede ver cómo se extrae la imagen (descarga) desde Docker Hub, se crea un contenedor y luego se inicia el contenedor.

También se asignan algunas reservas de CPU y memoria, que se pueden ajustar si es necesario. Se muestra una dirección IP pública, junto con cierta información sobre el contenedor, como el estado de aprovisionamiento, el tipo de SO y la directiva de reinicio.

- 4 Para abrir el sitio web básico que se ejecuta en el contenedor, puede consultar solo la dirección IP pública asignada:

```
az container show \
--resource-group azuremolchapter19 \
--name azuremol \
--query ipAddress.ip \
--output tsv
```

- 5 Abra la dirección IP pública de su instancia de contenedor en un navegador web. La pizzería básica debe mostrarse, como se muestra en la figura 19.7.

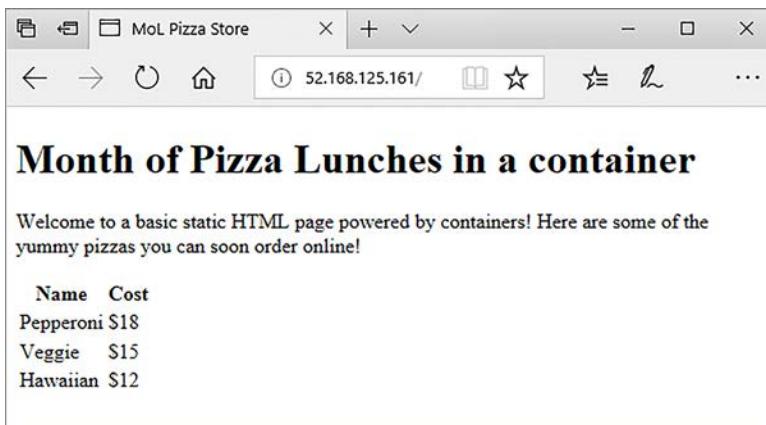


Figura 19.7 Al crear una instancia de contenedor, el sitio web de la pizzería se ejecuta sin ninguna configuración adicional. Toda la configuración y contenido se incluyen dentro de la imagen del contenedor. Este ejercicio rápido pone de manifiesto la portabilidad y el poder de los contenedores: cuando se ha preparado la imagen del contenedor, la aplicación queda activa y en funcionamiento tan pronto se implemente una nueva instancia de contenedor.

Examinemos la imagen del contenedor. No quiero alejarme demasiado de Docker y cómo desarrollar imágenes de contenedores, pero es importante entender de dónde viene esta imagen y cómo se ejecuta el sitio web sin ninguna configuración adicional.

La imagen se desarrolla a partir de una definición de configuración llamada *Dockerfile*. En Dockerfile, se define qué es la plataforma base, cualquier configuración que se desee aplicar y cualquier comando que se ejecute o los archivos que se deseé copiar. Dockerfiles puede ser, y a menudo es, más complejo que el ejemplo siguiente, que se utilizó para desarrollar el contenedor de ejemplo azuremol:

```
FROM nginx:1.17.5  
  
EXPOSE 80:80  
  
COPY index.html /usr/share/nginx/html
```

Cuando se utilizó este Dockerfile para crear una imagen de contenedor de Docker, se utilizó NGINX como imagen de origen y se copió la página web de ejemplo. Luego, este contenedor se llevó a Docker Hub, un repositorio público en línea que Docker proporciona para compartir e implementar contenedores. Para implementar la instancia de contenedor, usted proporcionó iainfoulds/azuremol como la imagen del contenedor que se utilizará. Azure buscó en Docker Hub y encontró un repositorio llamado iainfoulds y, dentro de él, una imagen denominada azuremol.

Examinemos cada línea del Dockerfile:

- `FROM nginx:1.17.5`: en capítulos anteriores, creó una VM básica, conectada a ella con SSH, y luego instaló manualmente el servidor web nginx. En el ejemplo Dockerfile, todo eso se logra en una línea. Esta línea dice que el contenedor debe basarse en una imagen de contenedor existente preinstalada con NGINX. 1.17.5 es la versión de la imagen de contenedor NGINX pública para utilizar; corresponde a la más reciente en el momento de la redacción de este libro. Es una buena práctica incluir un número de versión específico. Si no incluye un número de versión, siempre se utiliza la última versión. Esto suena bien en teoría, pero las aplicaciones de microservicios pueden escalar a un número bastante grande de contenedores activos, por lo que para asegurarse de tener un entorno coherente, se quiere controlar el número de versión exacto de cada componente en uso.
- `EXPOSE 80:80`: para permitir el acceso a la VM en capítulos anteriores, creó una regla NSG que permitía el puerto 80. En Dockerfile, esta línea le indica al contenedor que abra el puerto 80 y lo asigne al puerto interno 80. Cuando creó la instancia de contenedor con `az container create`, también especificó que la plataforma Azure debería permitir el tráfico con `--ports 80`. Esa es toda la red virtual en la que tiene que pensar.
- `COPY index.html /usr/share/nginx/html`: la parte final es llevar su aplicación al contenedor. En capítulos anteriores, utilizó Git para obtener la página web de ejemplo de la pizzería y luego llevó eso a su aplicación web. Con Dockerfile, usted `COPY` (copia) el archivo `index.html` al directorio local `/usr/share/nginx/html` del contenedor. ¡Listo!

Para sus propios escenarios, puede definir Dockerfile que utiliza una imagen base diferente, como Node.js o Python. A continuación, instale las bibliotecas o paquetes de soporte adicionales requeridos, extraiga el código de la aplicación del control de código fuente, como GitHub, e implemente la aplicación. Este Dockerfile se utilizaría para crear imágenes de contenedor que luego se almacenan en un registro de contenedor privado, no un repositorio público de Docker Hub como el del ejemplo.

Azure Container Registry

Puede que piense que Docker Hub suena muy bien. ¿Azure tiene algo maravilloso? Así es. Debido a que es necesario crear Dockerfile y desarrollar una imagen de contenedor, lamentablemente no es un ejercicio de dos minutos, y hay mucho que abordar en este capítulo. Puede integrar con facilidad Azure Container Registry (ACR) y AKS, de modo que ambos servicios funcionan bien juntos. Puede crear sus propias imágenes de Dockerfile en Cloud Shell; sin embargo, lo animo a explorar esto si dispone de tiempo. Azure Container Registry (ACR) es la ruta que elegiría para almacenar mis imágenes de contenedor, por un par de razones:

- Se trata de un registro privado para sus imágenes de contenedor, por lo que no es necesario preocuparse por el acceso potencial no deseado a sus archivos de aplicación y configuración. Puede aplicar los mismos mecanismos de RBAC que analizamos en el capítulo 6. RBAC le ayuda a limitar y auditar quién tiene acceso a sus imágenes.
- Almacenar las imágenes de su contenedor en un registro en Azure significa que sus imágenes están allí en los mismos centros de datos que la infraestructura utilizada para ejecutar sus instancias de contenedor o clústeres (que examinamos en la sección 19.4.1). Aunque las imágenes de contenedor deben ser relativamente pequeñas, a menudo solo decenas de MB de tamaño, que pueden sumar si sigue descargando esas imágenes de un registro remoto.

ACR también ofrece opciones de replicación y redundancia integradas que puede utilizar para colocar sus contenedores cerca de dónde se implementen y ejecutarlos para que los usuarios accedan. Esta localidad de región es similar a cómo utilizó la replicación global de Cosmos DB en el capítulo 10 para hacer que esos milisegundos cuenten y proporcionar a sus clientes el tiempo de acceso más rápido posible a sus aplicaciones.

Si todo esto suena emocionante, eche un vistazo a la página de inicio rápido de ACR para ponerse en funcionamiento con su propio repositorio privado en pocos minutos: <http://mng.bz/04rj>.

19.4 Azure Kubernetes Service

Ejecutar una instancia de contenedor única es genial, pero eso no le da mucha redundancia o capacidad de escalar. ¿Recuerda cómo pasamos capítulos completos antes en el libro hablando sobre cómo ejecutar múltiples instancias de su aplicación, equilibrio de carga y escalarlos automáticamente? ¿No sería genial hacer lo mismo con los contenedores? Ahí es donde se necesita un orquestador de contenedor.

Como su nombre lo indica, un *orquestador de contenedor* administra las instancias de contenedor, monitorea su estado y puede escalar según sea necesario. Los orquestadores pueden, y a menudo lo hacen, manipular mucho más, pero en un nivel alto, un enfoque principal está en manipular todas las partes móviles implicadas en la ejecución de una aplicación altamente disponible, escalable, basada en contenedores. Hay algunos orquestadores de contenedores, como Docker Swarm y el sistema operativo de nube distribuida (DC/OS), pero uno ha subido por encima del resto para convertirse en el orquestador listo de preferencia: Kubernetes.

Kubernetes comenzó como un proyecto de open source liderado por Google y que surgió de la herramienta de orquestación de contenedores internos de la empresa. Ampliamente aceptado por la comunidad de open source, Kubernetes es uno de los proyectos de open source más grandes y de más rápido crecimiento en GitHub. Muchas

empresas de tecnología de gran tamaño, incluidas Red Hat, IBM y Microsoft, contribuyen al proyecto Kubernetes básico.

En esta sección, vamos a tomar la misma aplicación web de ejemplo del ejercicio anterior con ACI para ejecutar una implementación redundante y escalable en Kubernetes. Terminará con algunos componentes, como se muestra en la figura 19.8.

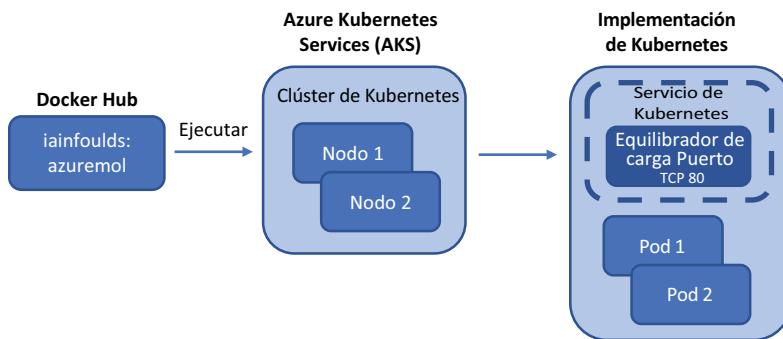


Figura 19.8 El contenedor de ejemplo de Docker Hub se ejecuta en un clúster Kubernetes de dos nodos que se crea en AKS. La implementación de Kubernetes contiene dos pods lógicos, uno en cada nodo de clúster, con una instancia de contenedor que se ejecuta dentro de cada pod. A continuación, expone un equilibrador de carga público para permitir que su aplicación web se vea en línea.

19.4.1 Creación de un clúster con Azure Kubernetes Services

En el capítulo 9, analizamos cómo los conjuntos de escala de máquinas virtuales reducen la complejidad de la implementación y configuración de la infraestructura subyacente. Usted indica cuántas instancias de VM desea en un conjunto de escala, y el resto de la red, el almacenamiento de información y la configuración se implementan para usted. AKS funciona de manera muy similar para ofrecer un clúster Kubernetes resiliente y escalable, con administración manipulada por la plataforma Azure. Los conjuntos de escalado pueden utilizarse para las máquinas virtuales subyacentes que se ejecutan en el clúster AKS, y esas máquinas virtuales pueden distribuirse por las zonas de disponibilidad. Se utilizan equilibradores de carga de Azure, también redundantes por zona. Básicamente, AKS reúne varios de los componentes de la infraestructura y los procedimientos recomendados que ha aprendido hasta ahora en este libro.

Pruébelo ahora

Para ver la información en el clúster de AKS, complete los pasos siguientes:

- 1 Abra Azure Portal y seleccione el icono de Cloud Shell en el menú superior.
- 2 Anteriormente en el capítulo, se creó un clúster Kubernetes. El proceso tardó unos minutos, pero espero que esté listo ahora. Mire el estado del clúster de la siguiente manera:

```
az aks show \
--resource-group azuremolchapter19 \
--name azuremol
```

El provisioningState cerca del final debe informar Succeeded.

- 3 Si el clúster está listo, obtenga un archivo de credenciales que le permita utilizar las herramientas de línea de comandos de Kubernetes para autenticar y administrar recursos:

```
az aks get-credentials \
--resource-group azuremolchapter19 \
--name azuremol
```

Eso es todo lo que se necesita para que Kubernetes funcione en Azure. Puede que esté preguntándose, "¿no puedo simplemente construir mi propio clúster con VM o conjuntos de escala, e instalar de forma manual el mismo Docker y componentes Kubernetes?". Claro que puede. El paralelo es el enfoque IaaS y PaaS de VM frente a aplicaciones web. El enfoque de aplicación web ofrece muchos beneficios: solo se preocupa por las opciones de configuración de alto nivel, y luego sube el código de su aplicación. Un clúster administrado de Kubernetes, ofrecido por AKS, reduce el nivel de complejidad y administración: su enfoque se convierte en sus aplicaciones y en la experiencia de sus clientes.

De la misma manera que puede elegir VM sobre aplicaciones web, puede optar por implementar su propio clúster Kubernetes en lugar de utilizar AKS. Está bien, ambos enfoques terminan usando los mismos componentes de servicios Azure. Las máquinas virtuales, los conjuntos de escala, los equilibradores de carga y los NSG son todos los temas que aprendió en capítulos anteriores, y todos están todavía presentes con los grupos AKS, aunque se abstraen. Desde una perspectiva de planificación y solución de problemas, debe tener las habilidades para entender lo que está sucediendo por debajo para hacer que funcione la oferta de Kubernetes administrada. Su nivel de comodidad, y el tiempo que desee para administrar la infraestructura, lo ayudarán a guiar su proceso de toma de decisiones mientras desarrolla una nueva aplicación en torno a los contenedores en Azure.

19.4.2 Ejecución de un sitio web básico en Kubernetes

Creó un clúster Kubernetes en la sección 19.4.1, pero no hay ninguna aplicación ejecutándose. ¡Vamos a cambiar eso! Necesita crear la implementación de Kubernetes que vio con anterioridad en la figura 19.8; vea la figura 19.9.

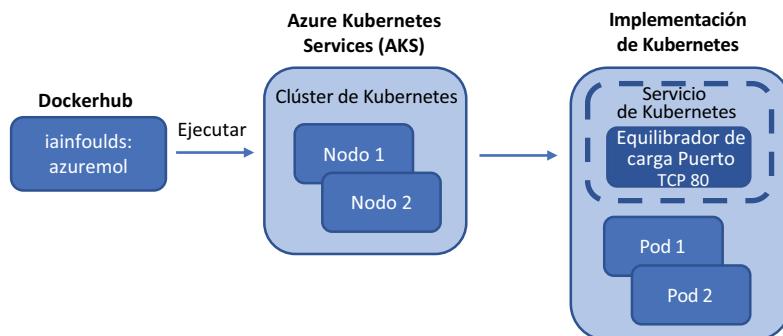


Figura 19.9 Con el clúster Kubernetes creado en AKS, puede crear una implementación de Kubernetes y ejecutar la aplicación. El contenedor se ejecuta a través de ambos nodos, con un pod lógico en cada nodo; es necesario crear un servicio Kubernetes que exponga un equilibrador de carga público para enrutar el tráfico a la aplicación.

Pruébelo ahora

Para implementar una aplicación en el clúster de Kubernetes, complete los pasos siguientes:

- 1 Interactúe con un clúster de Kubernetes mediante una utilidad de línea de comandos denominada kubectl. Utilice la misma imagen de contenedor iainfoulds/azuremol del Docker Hub que ejecutó como instancia de contenedor:

```
kubectl run azuremol \
--image=docker.io/iainfoulds/azuremol:latest \
--port=80
```

Puede tardar un minuto en descargar la imagen del contenedor de Docker Hub e iniciar la aplicación en Kubernetes. La aplicación se ejecuta en un *pod*: una construcción lógica en Kubernetes que aloja cada contenedor.

- 2 Los pods pueden contener componentes adicionales de ayudante, pero por ahora, supervise el estado de su contenedor observando el pod:

```
kubectl get pods --watch
```

Incluso cuando el estado del pod se informa como `Running`, no podrá acceder a la aplicación. La instancia de contenedor que creó con anterioridad podría enrutar el tráfico sobre una dirección IP pública directamente a esa instancia, pero ¿qué cree que se necesita para que un clúster Kubernetes enrute el tráfico a los contenedores? Si pensó en un equilibrador de carga, ¡felicitaciones! En este momento, solo tiene un pod: una instancia de contenedor único. Escalará el número de pods en el laboratorio de fin de capítulo, y para que funcione, se necesita una manera de enrutar el tráfico a varias instancias. Entonces, vamos a decirle a Kubernetes que use un equilibrador de carga.

Aquí es donde la integración entre Kubernetes y Azure se torna fantástica. Cuando le dice a Kubernetes que quiere crear un equilibrador de carga para sus contenedores, por debajo, Kubernetes vuelve a la plataforma Azure y crea un equilibrador de carga de Azure. Este equilibrador de carga de Azure es como el que aprendió en el capítulo 8. Hay grupos de IP frontend y back-end, y reglas de equilibrio de carga, y puede configurar sondeos de estado. A medida que su implementación de Kubernetes aumenta o disminuye, el equilibrador de carga se actualiza automáticamente según sea necesario.

Pruébelo ahora

Para exponer su aplicación a Internet, complete los siguientes pasos:

- 1 Dígale a Kubernetes que desea utilizar un equilibrador de carga y agregue una regla para distribuir el tráfico en el puerto 80:

```
kubectl expose deployment/azuremol \
--type="LoadBalancer" \
--port 80
```

- 2 Como antes, observe el estado de su implementación de servicios:

```
kubectl get service azuremol --watch
```

Cuando se asigne la dirección IP pública externa, el equilibrador de carga de Azure terminó de implementarse y el clúster y los nodos Kubernetes están conectados.

- 3 Abra la dirección IP pública de su servicio en un navegador web para ver cómo se ejecuta la aplicación web.

Las implementaciones de aplicaciones en Kubernetes a menudo están mucho más involucradas que este ejemplo básico. Normalmente se define un manifiesto de servicio, similar a una plantilla del administrador de recursos, que define todas las características de la aplicación. Estas propiedades pueden incluir el número de instancias de la aplicación que se ejecutará, cualquier almacenamiento para adjuntar, métodos de equilibrio de carga y puertos de red para utilizar, etc. En el mundo real, ni siquiera se hace esto manualmente; un sistema CI/CD como Azure DevOps o Jenkins automatiza las implementaciones de aplicaciones y los servicios directamente dentro del clúster AKS. Lo bueno de AKS es que no tiene que preocuparse por la instalación y configuración de Kubernetes. Al igual que con otros servicios PaaS, como aplicaciones web y Cosmos DB, usted lleva sus aplicaciones y deja que la plataforma Azure se encargue de la infraestructura subyacente y la redundancia.

Mantenerlo limpio y ordenado

Recuerde limpiar y eliminar sus grupos de recursos para que no acabe consumiendo un montón de sus créditos Azure gratis. A medida que empieza a explorar los contenedores, se vuelve aún más importante prestar atención a qué recursos de Azure deja activados. Una sola aplicación web no cuesta mucho, pero un clúster AKS de cinco nodos y unas pocas instancias de contenedor con imágenes de Azure Container Registry georeplicadas sí pueden.

Las instancias de ACI se cobran por segundo, y el costo se suma rápidamente si se dejan en funcionamiento durante días o semanas. Un clúster AKS ejecuta una VM para cada nodo, así que si escala y ejecuta muchas VM en su clúster, está pagando por una VM para cada nodo.

No hay ningún cargo por el número de contenedores que cada uno de los nodos AKS se ejecuta, pero como con cualquier VM, un nodo AKS se vuelve costoso cuando se deja en funcionamiento. Lo bueno de Kubernetes es que puede exportar sus configuraciones de servicio (la definición de sus pods, los equilibradores de carga, el autoescalado, etc.) para implementarlas en otros lugares. A medida que desarrolla y prueba sus aplicaciones, no necesita dejar un clúster de AKS en ejecución, puede implementar un clúster según sea necesario e implementar su servicio desde una configuración anterior.

Los clusters de AKS pueden escalar hacia arriba y hacia abajo, como verá en el ejercicio de laboratorio del final del capítulo. También puede configurar el escalado automático que hace este escalado por usted en función de la carga. Es el mismo tipo de escalado automático que vimos en el capítulo 9 para conjuntos de escalado y aplicaciones web. ¿Comienza a ver que todo se une en Azure?

Este capítulo también ha sido una introducción a velocidad warp para los contenedores y Kubernetes, así que no se preocupe si se siente un poco abrumado en este momento. Manning tiene varios libros excelentes, como *Aprenda Docker en un mes de almuerzos*,

(continuación)

por Elton Stoneman (<https://livebook.manning.com/book/learn-docker-in-a-month-of-lunches>) y *Kubernetes en acción*, de Marko Luksa (<https://livebook.manning.com/book/kubernetes-in-action>), que pueden ayudarle a profundizar más en Docker, el desarrollo de aplicaciones de microservicios y Kubernetes. Reviselos si este capítulo suena emocionante y quiere explorar más.

Los ejemplos de este capítulo utilizaron VM de Linux para los nodos del cluster de AKS y luego ejecutaron contenedores Linux para NGINX. Los contenedores son un poco complicados, ya que se pueden ejecutar contenedores Linux solo en nodos Linux, por ejemplo. Como aprendiste al principio del capítulo, los contenedores comparten el sistema operativo huésped y el kernel. Por lo tanto, no puede ejecutar contenedores de Windows en un nodo de Linux. En general, tampoco puede ejecutar contenedores de Linux en un nodo de Windows. Hay algunos trucos técnicos interesantes, pero en general, el contenedor y el sistema operativo del nodo subyacente deben coincidir.

Lo bueno de AKS es que puede ejecutar tanto nodos Linux como Windows, por lo que puede ejecutar contenedores Linux y Windows. Es necesario prestar un poco de atención a cómo se programan estos diferentes contenedores en los diferentes sistemas operativos de los nodos, pero este enfoque amplía enormemente las aplicaciones y servicios que se pueden ejecutar en AKS.

19.5 Laboratorio: Escalado de sus implementaciones de Kubernetes

El ejemplo básico de este capítulo creó un clúster Kubernetes de dos nodos y un único pod que ejecuta su sitio web. En este laboratorio, explore cómo puede escalar el clúster y el número de instancias de contenedor. Este ejemplo es básico, pero cuantos más nodos tenga, más instancias de contenedor podrá ejecutar, lo que es especialmente útil cuantas más aplicaciones necesite ejecutar en su clúster.

- 1 Puede ver cuántos nodos hay en su clúster Kubernetes con `kubectl get nodes`. Escale su clúster a tres nodos:

```
az aks scale \
--resource-group azuremolchapter19 \
--name azuremol \
--node-count 3
```

Se tarda un minuto o dos en escalar y agregar el nodo nuevo.

- 2 Utilice `kubectl` de nuevo para ver el estado de sus nodos. Cuando se amplía un nodo, Kubernetes no crea instancias de contenedor adicionales para sus aplicaciones de forma automática, por lo que no se obtiene ningún beneficio inmediato de los recursos informáticos adicionales que proporciona el nuevo nodo.
- 3 Vea su implementación actual con `kubectl get deployment azuremol`. Solo se creó una instancia antes. Esta aplicación de ejemplo no está aprovechando al máximo el nuevo nodo que agregó al cluster en el paso 1. Escale hasta cinco instancias o *réplicas*:

```
kubectl scale deployment azuremol --replicas 5
```

- 4 Utilice kubectl de nuevo para examinar la implementación. Mire los pods, las instancias del contenedor en ejecución con kubectl get pods. En cuestión de segundos, todas esas réplicas adicionales se iniciaron y se conectaron al equilibrador de carga.
- 5 Utilice kubectl get pods -o wide para ver en qué nodos se ejecutan los pods. Observe el último número en el nombre del nodo, que indica qué nodo del conjunto de escalado se utiliza. Los pods deben distribuirse en todos los nodos del clúster. Como otras aplicaciones escalarían el número de contenedores de forma similar, se puede empezar a maximizar el uso de los recursos informáticos en todos los nodos del clúster.

Azure y la Internet de las Cosas

Para mí, una de las áreas más emocionantes de la tecnología en los últimos años es la Internet de las Cosas (IoT). No creo que un lavavajillas o un refrigerador deban estar conectados a Internet aún, y hay preocupaciones válidas respecto de la privacidad sobre un televisor o un dispositivo de audio que están permanentemente conectados a Internet y siempre escuchando el sonido de su voz para emitir un comando. Sin embargo, hay muchas aplicaciones prácticas para dispositivos de IoT. Usted podría tener un informe de los equipos de fabricación sobre su estado, generar alertas de mantenimiento y permitir que los operadores comprendan su eficacia en varias fábricas en todo el mundo. Una empresa de camiones podría analizar los datos de telemetría de sus vehículos en relación con las cargas que se transportan y los tiempos de conducción promedio, y podría redireccionar la ruta de los conductores en forma inteligente según sea necesario. Las empresas de envío podrían hacer un seguimiento de cada contenedor y ayudar a sus clientes a administrar mejor su cadena de suministro al saber dónde se encuentran sus recursos.

En Azure, puede integrar muchos dispositivos de IoT con una gama de servicios. Azure Web Apps puede proporcionar un frontend para la visualización de sus datos, el almacenamiento se puede utilizar para registrar datos de dispositivos y las características sin servidor como Azure Logic Apps (que se explican en el siguiente capítulo final) pueden procesar los datos recibidos.

En este capítulo, analizaremos qué es la IoT y cómo utilizar Azure IoT Hub para administrar centralmente y recopilar datos de dispositivos. A continuación, verá cómo utilizar una aplicación web de Azure para ver datos en tiempo real desde un dispositivo de IoT.

20.1 ¿Qué es la Internet de las Cosas?

El interés en la IoT ha crecido considerablemente en los últimos años, pero es un término vago que se puede aplicar a muchos escenarios. En un nivel básico, IoT es un enfoque en el que muchos dispositivos interconectados, normalmente pequeños

dispositivos electrónicos de bajo costo, vuelven a conectarse a sistemas y aplicaciones centrales. En general, los dispositivos conectados brindan información que recopilan de las entradas o sensores conectados. Esta información luego puede ser procesada por un sistema central (probablemente con IA o ML, como se explica en el capítulo 17) y llevar a cabo las acciones apropiadas. La figura 20.1 muestra un enfoque de alto nivel a IoT.

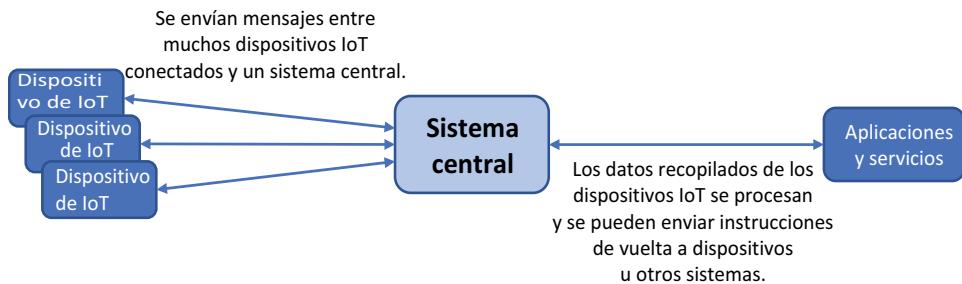


Figura 20.1 Se envían mensajes entre muchos dispositivos de IoT conectados y un sistema central. Sus aplicaciones y servicios pueden procesar los datos recibidos y enviar instrucciones al dispositivo para realizar acciones adicionales en respuesta a sus datos recopilados.

Algunos ejemplos de IoT en acción incluyen lo siguiente:

- *Garaje de estacionamiento*: un pequeño sensor por encima de cada zona de estacionamiento detecta si hay un vehículo estacionado allí. Una luz sobre cada zona puede encenderse de color verde si la zona está vacía o de color rojo si está ocupada. Los conductores que entran en el garaje de estacionamiento pueden ver paneles de información en tiempo real en cada piso que les permiten saber cuántos lugares de estacionamiento abierto hay. Las luces rojas y verdes sobre cada zona ayudan a los conductores a determinar rápidamente la ubicación de los puntos abiertos mientras conducen por cada pasillo.
- *Fábrica*: la maquinaria en un piso de la fábrica puede brindar información sobre el rendimiento operativo, los niveles de consumibles y las necesidades de mantenimiento. Un sistema central puede entonces programar a un técnico de mantenimiento para reparar proactivamente los equipos o reabastecer los consumibles, lo que reduce el tiempo de inactividad en la línea de producción. Cuando se combinan con IA y ML, se pueden predecir los horarios de mantenimiento, y se puede entregar la cantidad correcta de suministros o materias primas justo antes de que se necesiten en la producción.
- *Transporte*: los autobuses o trenes de transporte público pueden incluir sensores GPS que informen sobre la ubicación y la velocidad. También se puede recopilar información sobre los boletos para informar cuántas personas se están transportando. Los paneles de información para pasajeros en una estación de tren o terminal de autobuses pueden proporcionar información en tiempo real sobre cuándo llegará cada vehículo. Cuando se combina esta tecnología con IA y ML, los pasajeros en espera pueden recibir sugerencias de rutas alternativas basadas en condiciones de tráfico, retrasos o un gran volumen de pasajeros.

A menudo, IoT trabaja junto con otras aplicaciones y servicios. Los escenarios de la fábrica y del transporte podrían utilizar IA y ML para informar mejor la de la producción o para hacer sugerencias a los pasajeros. Las aplicaciones web pueden

utilizar la información recibida de los dispositivos de IoT para proporcionar acceso desde dispositivos móviles o generar alertas y notificaciones. Los datos recibidos de los dispositivos de IoT se pueden registrar en un sistema de bases de datos como Azure Cosmos DB y luego son procesados por aplicaciones de inteligencia empresarial y generan informes.

Las ideas hacia el futuro alrededor de IoT incluyen cosas como su refrigerador que detecta los niveles de alimentos y genera una lista de compras o incluso pedir comida a una tienda de comestibles local. Su automóvil podría brindar datos a la concesionaria, la que podría tener listas las piezas o consumibles necesarios cuando usted lleva el vehículo para mantenimiento. O ¿qué pasa si, cuando el despertador suena para despertarlo por la mañana, su cafetera se enciende y se prepara para el desayuno?

Una gran área de preocupación con IoT es la seguridad del dispositivo. Con tantos dispositivos fuera de su infraestructura de red principal y a menudo conectados a la Internet pública, poder aprovisionar, mantener y actualizar esos dispositivos es un desafío. Muchos dispositivos de IoT son de baja potencia, electrónica simple que pueden no tener las capacidades de almacenamiento o de procesamiento para actualizarse con actualizaciones de aplicaciones y de seguridad de la forma en que lo hace una computadora de escritorio tradicional o un equipo portátil. No es suficiente implementar muchos dispositivos de IoT, en especial dispositivos de nivel de consumidor, sin un plan para protegerlos correctamente y proporcionar actualizaciones y mantenimiento.

Estas preocupaciones de seguridad no deberían impedirle desarrollar aplicaciones y servicios que utilicen dispositivos de IoT. La IoT aporta un nuevo conjunto de desafíos para el mantenimiento de dispositivos tradicionales, pero hay soluciones que le permiten aprovisionar y mantener los dispositivos de forma centralizada, así como la comunicación segura del dispositivo.

Por ahora, estoy seguro de que habrá adivinado que Azure tiene una solución de IoT como esa. Ofrece un conjunto de servicios de IoT. Veamos cómo puede explorar la IoT con Azure.

Aceleración de las implementaciones de Azure IoT

Este capítulo se centra en Azure IoT Hub, un servicio que le permite aprovisionar y conectar dispositivos de IoT para desarrollar sus propias soluciones. Puede definir cómo se conectan esos dispositivos de IoT, qué usuarios o aplicaciones pueden acceder a sus datos y asegurar la conectividad. El modo para desarrollar e implementar la infraestructura de la aplicación para conectar todo depende de usted.

Los aceleradores de soluciones Azure IoT son escenarios clave preconfigurados, como supervisión remoto de dispositivos o una fábrica conectada. Los aceleradores implementan servicios comunes de Azure como IoT Hub, Web Apps, Cosmos DB y almacenamiento, y ejecutan una aplicación de ejemplo que integra todos estos servicios diferentes.

Usted todavía necesita personalizar la aplicación para su propio entorno, dispositivos de IoT en uso y los datos que se recopilarán y controlarán, pero los aceleradores de soluciones de IoT le brindan un gran marco para empezar. Mientras que IoT Hub crea una manera para que usted conecte los dispositivos de IoT a Azure y luego le permite implementar servicios adicionales que necesita, los aceleradores de soluciones de IoT implementan soluciones preconfiguradas que utilizan los servicios de Azure más comunes que usaría.

Si le interesa la IoT después de este capítulo y quiere obtener más información, los aceleradores de soluciones Azure IoT son una gran manera de ver las posibilidades de lo que Azure puede ofrecer. Como analizamos en este libro, Azure es mucho más que solo uno o dos servicios independientes. Puede implementar muchos servicios para proporcionar la mejor experiencia de aplicación posible para sus clientes.

20.2 Administración centralizada de dispositivos con Azure IoT Hub

Azure IoT Hub le permite administrar, actualizar y transmitir datos de forma centralizada desde dispositivos de IoT. Con este servicio, puede realizar acciones como la configuración de rutas de aplicación para datos recibidos desde dispositivos, aprovisionamiento y administración de certificados para asegurar la comunicación y control de la salud con diagnósticos y métricas Azure. Puede conectar sus dispositivos de IoT a otros servicios y aplicaciones Azure para permitirles enviar y recibir datos como parte de una solución más amplia. Al igual que con todas las cosas en Azure, el acceso puede ser controlado con RBAC, y los datos de diagnóstico pueden recopilarse de forma centralizada para la solución de problemas y control o alertas. La figura 20.2 describe el modo en que actúa un IoT Hub como lugar central para conectar dispositivos de IoT a los servicios y aplicaciones más amplios de Azure.

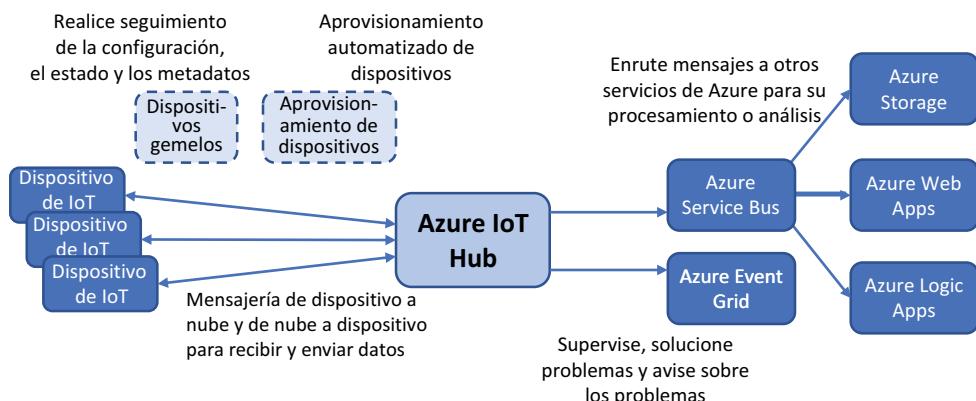


Figura 20.2 Con un IoT Hub, usted puede aprovisionar y administrar de forma centralizada varios dispositivos de IoT a escala. Existe una comunicación de dos vías entre los dispositivos y Azure para la lectura y escritura de datos. Puede procesar los datos recibidos de los dispositivos y enrutarlos a otros servicios Azure, como Web Apps y almacenamiento. Para controlar y solucionar problemas, puede enrutar información a Azure Event Grid, que analizaremos en el capítulo 21 y, a continuación, vincularse con otras soluciones de control.

Controla el acceso a IoT Hub con directivas de acceso compartido. Estas directivas son como cuentas de usuario y permisos. Existen directivas predeterminadas que permiten que los dispositivos y los servicios se conecten a IoT Hub, o que lean y escriban información desde el registro de dispositivos que realiza el seguimiento de dispositivos de IoT conectados y claves de seguridad. A cada directiva se le puede asignar uno o más de los siguientes permisos:

- Lectura del registro
- Escritura del registro

- Conexión de servicios
- Conexión de dispositivos

Las claves de acceso compartido son utilizadas por aplicaciones y servicios para conectarse a un IoT Hub. Al igual que con el almacenamiento de información (analizado en el capítulo 4), las claves de acceso compartido permiten definir cadenas de conexión para identificar el host, la directiva de acceso y la clave de acceso. Una cadena de conexión combina la clave de acceso, el tipo de directiva de acceso y el nombre de host de IoT Hub. Aquí hay una cadena de conexión de IoT Hub de ejemplo:

```
HostName=azurem01.azure-devices.net;SharedAccessKeyName=registryRead;  
➡ SharedAccessKey=6be2mXBVN9B+UkoPUMuwVDtR+7NZVBq+C7A1xCmQGAb=
```

Existen claves primarias y secundarias, que pueden rotarse y actualizarse por motivos de seguridad, al igual que la actualización periódica de contraseñas. Soluciones como Azure Key Vault (explicada en el capítulo 15) son una gran manera de hacer un seguimiento y almacenar estas claves para que las aplicaciones las obtengan cuando sea necesario. Este enfoque a la administración de claves significa que puede rotar con frecuencia las claves de acceso sin necesidad de actualizar también todo el código de la aplicación.

Los certificados digitales se pueden almacenar en un IoT Hub y aprovisionarse de forma automática a los dispositivos de IoT. Recuerde, los dispositivos de IoT a menudo están fuera de su infraestructura principal y pueden conectarse directamente a través de Internet sin ningún tipo de conexión de red segura como una VPN. Asegúrese de que todos los datos entre sus dispositivos y el IoT Hub estén cifrados mediante conexiones SSL/TLS. Azure Key Vault puede generar y almacenar certificados SSL que luego se agregan al IoT Hub. O puede utilizar una entidad de certificación para solicitar y emitir certificados. Lo importante es asegurarse de que toda la comunicación entre sus dispositivos de IoT y Azure esté cifrada. De lo contrario, es probable que reciba un error.

Las rutas de IoT Hub le permiten enviar datos desde dispositivos de IoT a otros servicios Azure. Puede definir criterios, como que el contenido del mensaje contenga una palabra clave o un valor determinado, y, a continuación, enrutar los mensajes para que se almacenen en Azure Storage o sean procesados por una aplicación web. En uno de los siguientes ejercicios, usted simulará un sensor de temperatura básico conectado a un dispositivo de IoT. Podría definir una ruta en IoT Hub para ver los datos entrantes y, si la temperatura registrada supera los 30 °F, enrutar los datos a una aplicación lógica para enviar una alerta por correo electrónico. Analizaremos el maravilloso mundo de la informática sin servidor y aplicaciones lógicas en el capítulo 21.

La vida con Edge

En este capítulo, nos centramos en Azure IoT Hub. Otro servicio, Azure IoT Edge, le permite ejecutar algunos servicios como Azure Functions y Stream Analytics en su entorno local. En lugar de tener todos los dispositivos IoT transmitiendo datos que se procesan centralmente en Azure, puede procesar los datos dentro de cada ubicación.

Azure IoT Edge ejecuta aplicaciones y servicios en contenedores (analizados en el capítulo 19). El uso de contenedores permite que IoT Edge sea portátil y coherente en el modo en que funciona en diferentes dispositivos y entornos. Se pueden implementar servicios Azure preconfigurados, o puede escribir sus propias aplicaciones y distribuirlas a ubicaciones de Edge.

El principal beneficio de IoT Edge es que descarga parte del procesamiento de datos y de las transferencias de datos de red. Si puede procesar datos localmente en IoT Edge, puede agrupar en lotes grandes fragmentos de datos y transmitirlos de nuevo a Azure. Las aplicaciones centrales pueden entonces incorporar información de otras ubicaciones de perímetro para que sea procesada por servicios como IA y ML.

Otro gran escenario para Azure IoT Edge son las ubicaciones remotas, que a menudo se encuentran en las industrias de petróleo y gas o de transporte, donde la conectividad a Internet puede no ser lo suficientemente confiable para la transmisión de todos los datos de dispositivos de IoT a Azure para el procesamiento central. IoT Edge permite que esas ubicaciones remotas continúen funcionando con cierta autonomía, incluso cuando no hay conexión a Internet.

Al planificar una infraestructura de aplicaciones que involucre dispositivos de IoT, examine cómo manipula las interrupciones de red y las conexiones de Internet deficientes. Si su entorno se basa en Internet, planifique conexiones de Internet y equipos redundantes para enrutar los datos. U observe IoT Edge para procesar de forma local los datos cuando no se puede hacer centralmente en Azure

Pruébelo ahora

Para empezar con IoT y crear un IoT Hub, complete los siguientes pasos:

- 1 Abra Azure Portal; inicie Cloud Shell; y cree un grupo de recursos como azuremolchapter20:

```
az group create --name azuremolchapter20 --location eastus
```
- 2 Ha trabajado mucho con la CLI de Azure en este libro, dado que los comandos de Cloud Shell y CLI le permiten crear y administrar recursos rápidamente. Como se mencionó en capítulos anteriores, la CLI de Azure también puede utilizar módulos adicionales, denominados *extensiones*. Estas extensiones agregan más funcionalidad y a menudo se actualizan fuera del ciclo de lanzamiento normal de Acure CLI principal. Azure IoT se está expandiendo rápidamente y está añadiendo nuevas características, por lo que los comandos principales para interactuar con IoT Hub provienen de una extensión de la CLI de Azure.

Para obtener la funcionalidad completa que necesita para estos ejercicios, instale la extensión CLI de Azure IoT:

```
az extension add --name azure-cli-iot-ext
```

- 3 Cree un IoT Hub y escriba un nombre, como azuremol. Para estos ejercicios, puede utilizar un IoT Hub de nivel gratis, f1:

```
az iot hub create \
--resource-group azuremolchapter20 \
--name azuremol \
--sku f1 \
--partition-count 2
```

NOTA Puede crear solo un hub de nivel gratis por suscripción, pero estos hubs son excelentes para probar la comunicación entre dispositivos e integrarse con

otros servicios Azure. El hub de nivel gratuito está actualmente limitado a 8000 mensajes al día y admite un máximo de 500 dispositivos conectados. Esto puede parecer mucho, pero según lo que esté haciendo, un solo dispositivo que envía un mensaje a IoT Hub aproximadamente cada 12 segundos alcanzará ese límite de 8000 mensajes.

Su IoT Hub está bastante vacío ahora. No hay mucho que se puede hacer con este sin uno o más dispositivos de IoT conectados. Un dispositivo común usado para IoT es el Raspberry PI. Se trata de un miniequipo de bajo costo que se puede conectar a redes Wi-Fi y utilizar los sensores comunes listos para usar para la temperatura, la humedad y la presión. También se puede utilizar para controlar pequeños motores, luces y temporizadores. No obstante, no es necesario salir corriendo y comprar una Raspberry Pi para trabajar con IoT Hub, puede simular uno en su navegador web.

20.3 Creación de un dispositivo Raspberry Pi simulado

Los dispositivos de IoT son geniales, pero hay una barrera de entrada en que necesita un dispositivo real para usar, ¿verdad? ¡No! Hay algunas maneras en las que puede simular un dispositivo de IoT con software. Este enfoque basado en software le permite centrarse en el desarrollo de su aplicación rápidamente y luego en la transición al hardware real. Todavía es necesario prestar atención a la forma en que el código se ejecuta en el hardware de IoT real, especialmente en dispositivos de baja potencia, ya que es posible que no tengan acceso a todas las bibliotecas necesarias, o incluso recursos de memoria, a las que tiene acceso la aplicación simulada.

Microsoft proporciona un simulador de Raspberry Pi gratis a través de GitHub en <https://azure-samples.github.io/raspberry-pi-web-simulator>. Una Raspberry Pi es estupenda para hacer pruebas, pero hay que tener cuidado cuando se utiliza un hardware barato como la Raspberry Pi en entornos de producción. Planifique cómo actualizar y administrar estos dispositivos. Los dispositivos de IoT dedicados, como Azure Sphere (<https://azure.microsoft.com/services/azure-sphere>), proporcionan opciones de seguridad y administración adicionales. Para este libro y en sus propias pruebas y aprendizaje, Raspberry Pi es una buena alternativa. En este simulador, sensor BME280 común que recopila lecturas de temperatura y humedad es simulado en software, junto con un LED simulado para mostrar cuando el dispositivo transmite datos al IoT Hub. No puede personalizar esto mucho, pero le permite ver cómo una aplicación básica Node.js puede ejecutarse en la Raspberry Pi, sondear los datos de un sensor, y enviarlos de vuelta a Azure.

NOTA Si cosas como Raspberry PI, los sensores electrónicos y de temperatura, y Node.js parecen intimidantes, no se preocupe. Al igual que con los capítulos de IA y ML, los contenedores, y Kubernetes, no vamos a profundizar demasiado en los dispositivos y la programación de IoT. Sin embargo, si usted siente que desea enchufar un soldador y profundizar en la electrónica al final de este capítulo, es más que bienvenido a hacerlo.

Antes de poder utilizar el simulador de Raspberry Pi, es necesario crear una asignación de dispositivo en Azure IoT Hub. Este proceso crea un ID de dispositivo único para que su IoT Hub entienda con qué dispositivo se está comunicando y cómo procesar los datos. En escenarios más complejos, podría aprovisionar configuraciones adicionales para el dispositivo e insertar certificados digitales. Para este ejercicio, solo tendrá que crear una identidad de dispositivo.

Pruébelo ahora

Para crear un dispositivo de IoT de Raspberry Pi simulado, complete los siguientes pasos:

- 1 En Azure Cloud Shell, cree una identidad de dispositivo en su IoT Hub, como `azuremol` y proporcione un nombre para el dispositivo, como `raspberrypi`:

```
az iot hub device-identity create \
--hub-name azuremol \
--device-id raspberrypi
```

- 2 ¿Recuerda las directivas de acceso compartido de la sección 20.2? Cada dispositivo de IoT también tiene su propia clave de acceso y cadena de conexión que se utilizan para identificarlo cuando se comunica de nuevo al IoT Hub. Esta característica clave de Azure IoT asegura los dispositivos y minimiza el riesgo de exposición si un dispositivo se ve comprometido.

Para utilizar el dispositivo con el simulador RaspberryPi, necesita la información para la cadena de conexión del dispositivo. Este identificador único incluye el nombre de host de su IoT Hub, el ID del dispositivo y una clave de acceso:

```
az iot hub device-identity show-connection-string \
--hub-name azuremol \
--device-id raspberrypi \
--output tsv
```

- 3 Copie el contenido de su cadena de conexión; lo necesitará en el paso 4. La salida es similar a lo siguiente:

```
HostName=azuremol.azure-devices.net;DeviceId=raspberrypi;
➡ SharedAccessKey=oXVvK40qYYI3M4u6ZLxoyR/PUKV7A7RF/JR9WcsRYSI=
```

- 4 Ahora viene la parte divertida. Abra el simulador de Raspberry PI en su navegador web: <https://azure-samples.github.io/raspberry-pi-web-simulator>. Busque en la sección de código a la derecha en el simulador. Alrededor de la línea 15, debería haber una variable `connectionString`, que ya le solicite *[Su cadena de conexión del dispositivo de IoT Hub]*. Copie y pegue la cadena de conexión del paso 3, como se muestra en la figura 20.3.
- 5 Seleccione el botón Ejecutar justo debajo de la ventana de código para iniciar el simulador.

Cada dos segundos, la ventana de la consola muestra un mensaje que muestra los datos enviados al IoT Hub. La luz LED roja en el diagrama del circuito también parpadea cuando esto ocurre, para simular el modo en que se pueden controlar las salidas conectadas a Raspberry Pi. El mensaje de salida en la ventana de la consola es similar al siguiente:

```
Sending message: {"messageId":1,"deviceId":"Raspberry Pi Web
➡ Client","temperature":24.207095037347923,
➡ "humidity":69.12946775681091}
```

¿De dónde provienen las lecturas de temperatura y humedad? Este dispositivo es un Raspberry Pi simulado y no hay un sensor BME280 real, de modo que la aplicación genera estos valores en software. Si observa el resto del código en la

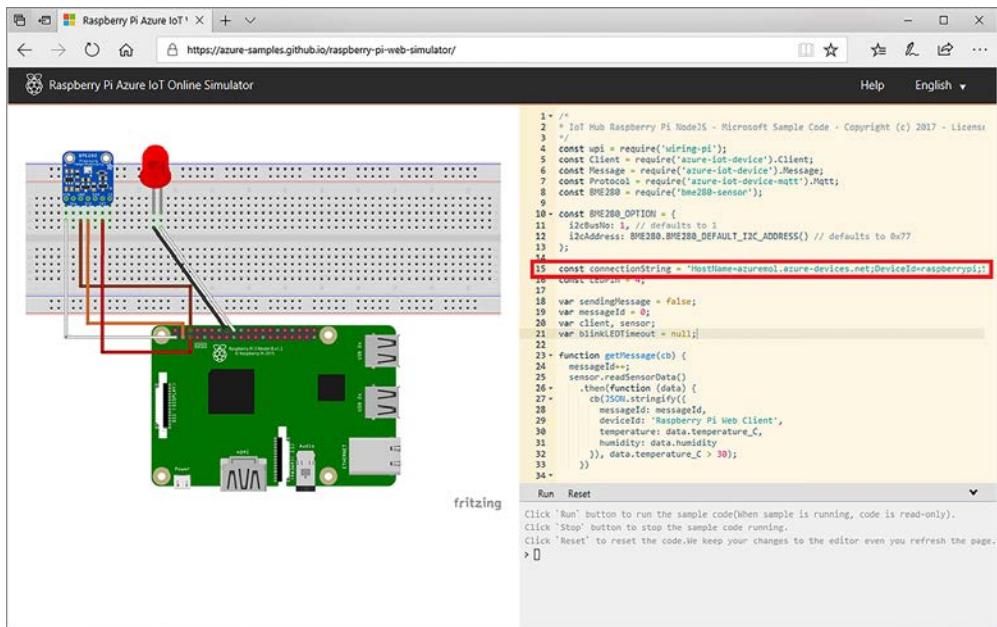


Figura 20.3 Copie y pegue la cadena de conexión para su dispositivo Azure IoT en el simulador Raspberry Pi. La variable connectionString se utiliza para conectarse para transmitir los datos del sensor simulado a Azure.

ventana del simulador, alrededor de la línea 99 la aplicación define el sensor. El simulador entonces replica el modo en que actuaría el sensor real y genera los datos devueltos del sensor a la aplicación. Este es un ejemplo básico, así que piense qué más puede leer aquí: revoluciones por minuto (RPM) de un motor, o coordenadas de GPS de un contenedor o camión de envío, etc. Aquí es donde hay un equilibrio entre la simulación de un dispositivo en software y el desarrollo de una aplicación funcional con el hardware real y los datos del sensor. En algún momento, usted necesita comprar o pedir prestado equipos si desea profundizar más en Azure IoT.

- 6 Para confirmar que el IoT Hub está recibiendo los mensajes de su dispositivo simulado, examine el estado de la cuota. Proporcione el nombre de su IoT Hub, como azuremol:

```
az iot hub show-quota-metrics --name azuremol
```

La salida es similar al siguiente ejemplo, que muestra que se han recibido 5 mensajes del máximo de 8000 mensajes totales por día y que hay un dispositivo conectado de un máximo de 500 dispositivos totales. Puede que tome unos minutos para que estas métricas se llenen, así que no se preocupe si no ve ningún dato inmediatamente:

```
[  
 {  
   "currentValue": 5,
```

```
    "maxValue": 8000,
    "name": "TotalMessages"
},
{
    "currentValue": 1,
    "maxValue": 500,
    "name": "TotalDeviceCount"
}
]
```

También puede buscar en Azure Portal: elija su grupo de recursos y, a continuación, seleccione su IoT Hub. En la página de información general, el uso del hub informa la cantidad de mensajes recibidos y de los dispositivos conectados. Una vez más, puede tomar un minuto o dos para que aparezcan los mensajes y se registren respecto de la cuota. Cualquier aplicación sería capaz de utilizar inmediatamente los mensajes recibidos, tal como lo vemos en la sección 20.4.

Problemas en el paraíso

Si no recibe ningún mensaje en su IoT Hub, compruebe la ventana de salida de su dispositivo Raspberry Pi simulado. Una de las primeras cosas que hace la aplicación es conectarse con Azure IoT Hub. Se muestra un error de conexión si la cadena de conexión es incorrecta. Asegúrese de copiar y pegar correctamente toda la cadena de conexión. La cadena de conexión comienza con `HostName`, y el último carácter en todas las claves de acceso es siempre un signo igual (=).

Si la ventana de salida informa un error, copie el texto del error en su motor de búsqueda favorito y busque un resultado que coincida. Asegúrese de que no ha cambiado ninguna de las otras líneas de código, lo que causaría un problema. Lo único que necesita cambiar en la ventana de código es la línea de la cadena de conexión.

Debido a que el dispositivo Raspberry Pi simulado se ejecuta en un navegador web, podría tener un problema de sitio web genérico. Intente actualizar la página, o acceda al simulador en un navegador diferente (<https://azure-samples.github.io/raspberry-pi-web-simulator>).

20.4 Transmisión de datos de Azure IoT Hub a las aplicaciones web de Azure

Un dispositivo que se conecta a un IoT Hub no sirve si no puede hacer nada con los datos. Aquí es donde puede comenzar a integrar muchos de los servicios y características que ha aprendido en este libro. ¿Desea transmitir a las tablas o colas de Azure Storage? Puedes hacer eso. ¿Procesar datos de dispositivos de IoT en VM o contenedores Azure? ¡Adelante! ¿Utilizar Azure Cosmos DB para replicar sus datos y, a continuación, acceder a estos con aplicaciones web Azure globalmente redundantes y Traffic Manager? ¡Por supuesto!

En el escenario de ejemplo, el IoT Hub es el mecanismo de conexión y el punto de entrada para sus dispositivos de IoT a Azure. El hub en sí no hace nada directamente con los datos. Existe un punto de conexión predeterminado para eventos, que es un sector grande para cualquier mensaje recibido del dispositivo de IoT. Su dispositivo Raspberry Pi simulado envía mensajes a IoT Hub, y estos mensajes llegan a este punto de conexión de eventos. En la figura 20.4, se muestra el flujo de mensajes desde dispositivos a través del IoT Hub a un punto de conexión.

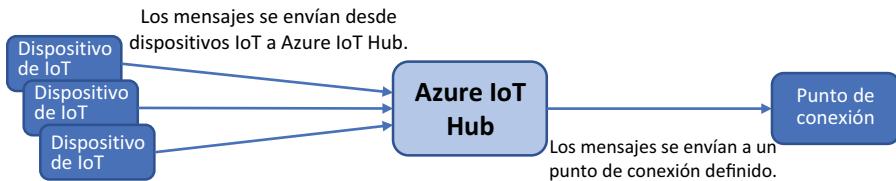


Figura 20.4 Un IoT Hub recibe mensajes de dispositivos de IoT conectados y envía los mensajes a un punto de conexión. Estos puntos de conexión pueden ser utilizados por otros servicios de Azure para consumir datos de los dispositivos de IoT. Existe un punto de conexión para eventos, que servicios como las aplicaciones web pueden leer.

Puede crear puntos de conexión personalizados que enruten los mensajes directamente a los servicios de Azure, como almacenamiento y Service Bus. En el capítulo 4, vimos las colas de Azure Storage para conocer una manera de pasar mensajes entre aplicaciones. Una plataforma de mensajería empresarial más sólida y escalable es Azure Service Bus. Se pueden agregar los mensajes al Service Bus, como los datos recibidos de los dispositivos de IoT, y otras aplicaciones pueden escuchar estos mensajes y responder en consecuencia.

Si no necesita la complejidad de leer mensajes de algo como un Service Bus, puede utilizar grupos de consumidores con el punto de conexión de eventos predeterminado. Un grupo de consumidores permite que los servicios como Azure Web Apps lean los datos del punto de conexión, como se muestra en la figura 20.5. Cada lectura de servicio de Azure IoT Hub debe tener su propio grupo de consumidores. Varios servicios, cada uno con su propio grupo de consumidores, pueden recibir los mismos mensajes y procesarlos según sea necesario.

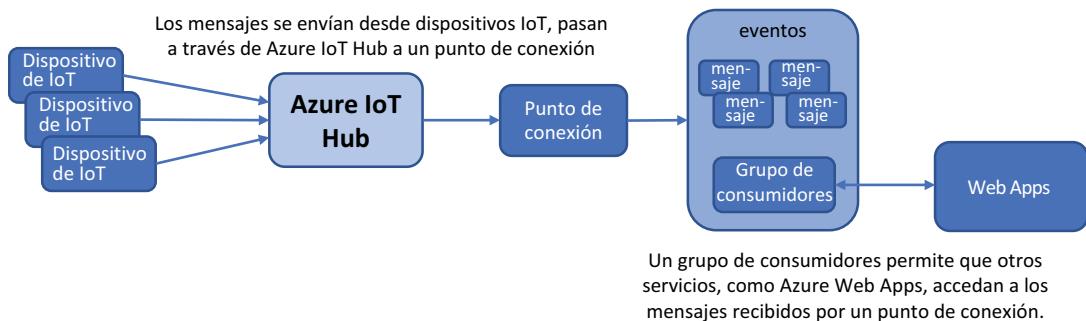


Figura 20.5 Los mensajes se envían desde dispositivos de IoT a IoT Hub, que luego dirige los mensajes a un punto de conexión. En cada punto de conexión, se pueden crear grupos de consumidores. Estos grupos de consumidores permiten que otros servicios de Azure accedan a los mensajes de dispositivo, a los que de otro modo no tendrían acceso. Con los grupos de consumidores, no es necesario utilizar colas de mensajes para permitir que las aplicaciones externas lean datos de dispositivos de IoT.

Vamos a crear una aplicación web de Azure que utiliza un grupo de consumidores para leer los datos del mensaje en tiempo real de su dispositivo Raspberry Pi simulado. Este ejemplo básico muestra cómo se pueden transmitir datos desde dispositivos de IoT y acceder a ellos desde aplicaciones web.

Pruébelo ahora

Para crear una aplicación web Azure que lea datos de dispositivos de IoT, complete los siguientes pasos:

- Cree un plan de Azure App Service para su aplicación web en Azure Cloud Shell y proporcione un nombre, como azuremol. Para estos ejercicios, el nivel gratis (f1) es lo suficientemente bueno y mantiene los costos bajos:

```
az appservice plan create \
--resource-group azuremolchapter20 \
--name azuremol \
--sku f1
```

- Cree su aplicación web. Proporcione un nombre, como molwebapp, y habilítelo para usarlo con Git de modo que pueda implementar la aplicación de ejemplo. Al igual que con otros recursos de Azure de acceso público, tiene que proporcionar su propio nombre único global

```
az webapp create \
--resource-group azuremolchapter20 \
--plan azuremol \
--name molwebapp \
--deployment-local-git
```

- Defina el grupo de consumidores para su IoT Hub, junto con algunas configuraciones de aplicaciones web. Estas configuraciones permiten que su aplicación web se conecte a su IoT Hub. En la figura 20.6, se muestra lo que desarrollará en los siguientes pasos.

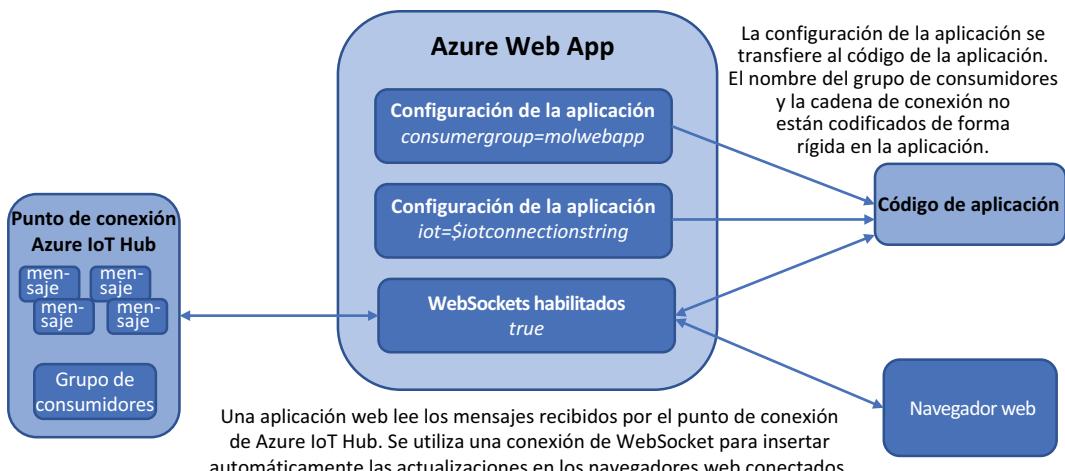


Figura 20.6 Para permitir que la aplicación web lea los datos de su dispositivo de IoT Raspberry Pi simulado, cree un grupo de consumidores en IoT Hub. A continuación, defina dos configuraciones de aplicaciones para su aplicación web que le permitan conectarse al grupo de consumidores. Para permitir que su navegador web reciba automáticamente el flujo de datos de Raspberry Pi a medida que se reciban nuevos datos, también habilitará una configuración para WebSockets.

- 4 Cree un grupo de consumidores que permita que su aplicación web acceda a los datos de eventos transmitidos desde su dispositivo de IoT. Proporcione su IoT Hub, como azuremol, y, a continuación, escriba un nombre para su grupo de consumidores como molwebapp. Asegúrese de utilizar su propio nombre en los siguientes pasos. Su grupo de consumidores se crea en el punto de conexión de eventos predeterminado:

```
az iot hub consumer-group create \
--hub-name azuremol \
--name molwebapp
```

- 5 Debe indicarle a su aplicación web cómo se llama el grupo de consumidores. Cree una configuración de aplicaciones para la aplicación web que utilice la aplicación de ejemplo que implemente al final del ejercicio. La configuración de la aplicación en aplicaciones web permite definir configuraciones específicas, como el nombre del grupo de consumidores y la cadena de conexión, sin que esos valores se codifiquen de forma rígida en su aplicación.

Proporcione el nombre del grupo de consumidores creado en el paso 4, como mol-webapp:

```
az webapp config appsettings set \
--resource-group azuremolchapter20 \
--name molwebapp \
--settings consumergroup=molwebapp
```

- 6 Para conectarse a su IoT Hub, su aplicación web necesita conocer la cadena de conexión para el Hub. Esta cadena de conexión es diferente a la que usted copió para su dispositivo Raspberry Pi simulado en el ejercicio anterior. Recuerde, hay una cadena de conexión para su IoT Hub, que utiliza directivas de acceso compartido para definir permisos de acceso; y hay una cadena de conexión para cada dispositivo de IoT. Su aplicación web necesita leer desde el grupo de consumidores del punto de conexión de IoT Hub, por lo que debe definir una cadena de conexión para el propio IoT Hub.

- 7 Obtenga la cadena de conexión de IoT Hub y asígnele una variable denominada iotconnectionstring, que se utiliza en el paso 8:

```
iotconnectionstring=$(az iot hub show-connection-string \
--hub-name azuremol \
--output tsv)
```

- 8 Cree otra configuración de la aplicación para la aplicación web, esta vez para la cadena de conexión de IoT Hub. La variable definida en el paso 7 se utiliza para permitir que la aplicación de ejemplo se conecte y lea los datos del dispositivo de IoT:

```
az webapp config appsettings set \
--resource-group azuremolchapter20 \
--name molwebapp \
--settings iot=$iotconnectionstring
```

- 9 Habilite WebSockets. Un *WebSocket* es un medio de comunicación bidireccional entre un navegador y un servidor. La aplicación de ejemplo actualiza automáticamente el navegador web con los datos recibidos desde el dispositivo

Raspberry Pi. Para realizar esta actualización automatizada, la aplicación utiliza WebSockets. A continuación, el servidor puede insertar datos en el navegador y hacer que se actualice automáticamente:

```
az webapp config set \
--resource-group azuremolchapter20 \
--name molwebapp \
--web-sockets-enabled
```

Vamos a hacer una pausa aquí para analizar lo que ha hecho hasta ahora. Ha trabajado con aplicaciones web en muchos de los capítulos anteriores, pero la configuración de la aplicación para la aplicación web y los WebSockets son nuevos. En la figura 20.7, se resume el modo en que se conectan su aplicación web e IoT Hub.

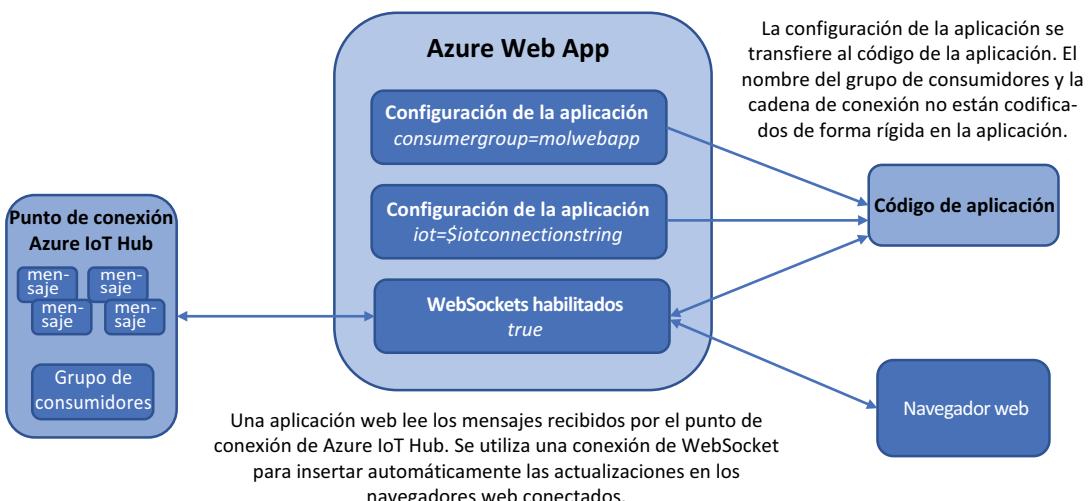


Figura 20.7 A medida que los mensajes se envían desde dispositivos de IoT, pasan a través de IoT Hub a un punto de conexión. Su código de aplicación lee en la configuración de la aplicación para la aplicación web que define la cadena de conexión de IoT Hub y el grupo de consumidores que se usarán. Una vez que la aplicación esté conectada a IoT Hub, el grupo de consumidores permite a las aplicaciones web leer los mensajes de dispositivo de IoT. Cada vez que se recibe un mensaje nuevo de un dispositivo de IoT, su aplicación web utiliza una conexión WebSocket con navegadores web que acceden a su sitio para insertar automáticamente actualizaciones. Esta conexión le permite ver datos en tiempo real transmitidos desde dispositivos de IoT, tales como la información de temperatura y humedad, desde su dispositivo Raspberry Pi simulado.

Ahora vamos a terminar el ejercicio e implementar la aplicación de ejemplo desde el repositorio de GitHub a su aplicación web. A continuación, puede abrir la aplicación web en su navegador y ver los datos en tiempo real transmitidos desde su Raspberry Pi simulado.

- 10 Si fuera necesario, clone el repositorio de ejemplos de GitHub en su Cloud Shell de la siguiente manera:

```
git clone https://github.com/fouldsy/azure-mol-samples-2nd-ed.git
```

- 11 Cambie al directorio para iniciar el capítulo 20:

```
cd azure-mol-samples-2nd-ed/20
```

- 12** Inicialice el repositorio Git y agregue la página web básica:

```
git init && git add . && git commit -m "Pizza"
```

- 13** Para cargar la aplicación de ejemplo, cree una conexión a su aplicación web. El siguiente comando obtiene el repositorio de la aplicación web y configura su repositorio Git de ejemplo local para conectarse con él:

```
git remote add molwebapp \
$(az webapp deployment source config-local-git \
--resource-group azuremolchapter20 \
--name molwebapp \
--output tsv)
```

En capítulos anteriores, tuvo que buscar esta dirección; pero por ahora espero que haya empezado a explorar qué más puede hacer la CLI de Azure y se haya dado cuenta de que gran parte de esta información se puede obtener rápidamente:

- 14** Inserte el sitio HTML de ejemplo en su aplicación web con el siguiente comando:

```
git push molwebapp master
```

- 15** Cuando se le solicite, ingrese la contraseña para el usuario de Git que creó y utilizó en capítulos anteriores (la cuenta creada en el capítulo 3).

Si no escribió su contraseña de Git en una nota rápida

Si olvidó la contraseña, puede restablecerla. Primero, obtenga el nombre de usuario de su cuenta de implementación de Git local:

```
az webapp deployment user show --query publishingUserName
```

Para restablecer la contraseña, introduzca el nombre de su cuenta desde el comando anterior y, a continuación, siga las instrucciones para configurar una nueva contraseña. El siguiente ejemplo restablece la contraseña de la cuenta de usuario denominada azuremol:

```
az webapp deployment user set --user-name azuremol
```

- 16** Visualice el nombre de host para su aplicación web y, a continuación, abra la dirección en un navegador web:

```
az webapp show \
--resource-group azuremolchapter20 \
--name molwebapp \
--query defaultHostName \
--output tsv
```

La primera vez que abra el sitio en su navegador web puede tardar unos segundos, ya que la aplicación web se conecta a IoT Hub, inicia la conexión de WebSocket y espera a que se reciba el primer mensaje de dispositivo. Cada dos segundos, el navegador web debe actualizarse automáticamente con los últimos datos simulados del dispositivo Raspberry Pi, como se muestra en la figura 20.8.

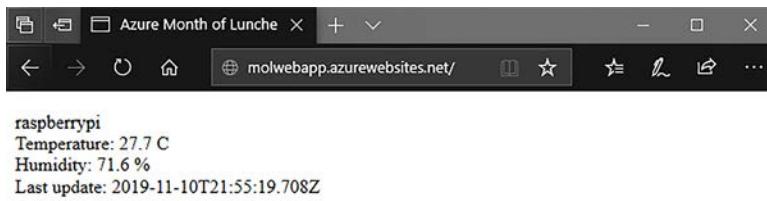


Figura 20.8 La aplicación de ejemplo utiliza una conexión WebSocket entre el navegador web y la aplicación web para actualizarse automáticamente cada dos segundos con los datos más recientes de su dispositivo Raspberry Pi simulado.

Si la instancia de su aplicación web no muestra ningún dato, asegúrese de que el dispositivo Raspberry Pi simulado siga ejecutándose. Si es necesario, inicie el dispositivo simulado y asegúrese de que se conecte a Azure IoT y envíe mensajes. Los datos deben comenzar a aparecer en la instancia de la aplicación web.

20.5 Revisión de componentes de Azure IoT

Espero que los ejercicios en este capítulo le hayan dado una idea de los servicios que están disponibles en Azure para las soluciones de IoT:

- *Azure IoT Hub* brinda una gran manera de aprovisionar, conectar y administrar muchos dispositivos de IoT y luego se integran con otros servicios de Azure.
- *Los aceleradores de soluciones de Azure IoT* proporcionan escenarios preconfigurados que integran automáticamente muchos servicios de Azure para proporcionar un entorno de aplicación completo.
- *Azure IoT Edge* le permite implementar servicios Azure en su entorno local para procesar los datos de los dispositivos de IoT sin la necesidad de transmitir todos los datos de forma centralizada a Azure.

Para analizar realmente los dispositivos Azure IoT e IoT en general, le recomiendo que compre un dispositivo Raspberry Pi básico o similar. Estos dispositivos son relativamente económicos, a menudo vienen con algunos sensores básicos o componentes eléctricos para probar diferentes ideas y le ofrecen una gran plataforma de aprendizaje cuando ve lo que es posible al integrar el hardware y el software. Solo recuerde las advertencias del capítulo 17 sobre IA y ML y la construcción de Skynet. Manning también tiene algunos libros excelentes, tales como *Building the Web of Things* (Creación de la Web de las Cosas), de Dominique D. Guinard y Vlad M. Trifa (<https://www.manning.com/books/building-the-web-of-things>) y *JavaScript on Things* (JavaScript en las Cosas), de Lyza Danger Gardner (<https://www.manning.com/books/javascript-on-things>), que profundizan sobre Raspberry Pi, los procedimientos recomendados de IoT, y la programación de JavaScript y Node.js en dispositivos IoT.

¿Recuerda que dije "siempre elimine los grupos de recursos"?

El procedimiento recomendado a lo largo de este libro ha sido eliminar sus grupos de recursos al final de cada capítulo. Este enfoque garantiza que no deje los servicios y las aplicaciones en uso que cuestan dinero cuando no los necesita.

(continuación)

Azure IoT le ofrece una gran plataforma para transmitir datos a Azure. En general, necesita procesar esos datos, no solo mostrarlos en una aplicación web como lo hizo en los ejercicios. El capítulo 21 analiza la informática sin servidor con los servicios de funciones y aplicaciones lógicas.

Para mostrar cómo estos servicios Azure funcionan bien juntos, no elimine el grupo de recursos y los servicios que implementó en este capítulo. Los utilizará de inmediato al comienzo del capítulo 21 para ver cómo puede tomar acciones basadas en los datos recibidos de sus dispositivos de IoT. Solo asegúrese de volver a su dispositivo Raspberry Pi simulado y seleccione el botón Detener; de lo contrario, el límite de 8000 mensajes se consumirá muy rápidamente.

20.6 Laboratorio: Exploración de casos prácticos para IoT

Este capítulo analizó muchas cosas nuevas, y sin un dispositivo de IoT real, está limitado respecto de lo que puede hacer. El Capítulo 21 se basa en Azure IoT Hub y Raspberry Pi simulado, por lo que no quiero configurar mucho más en este momento. Aquí algunas cosas que usted puede hacer para pensar más sobre IoT:

- 1 ¿En qué áreas puede pensar que los dispositivos de IoT podrían beneficiar a su negocio? Si no trabaja en un negocio en este momento, piense en la pizzería ficticia del *Mes de almuerzos de Azure*.
- 2 ¿Qué podría hacer para mejorar las cosas para los clientes con IoT?
- 3 ¿Utilizaría Azure IoT Edge? ¿Por qué sí o por qué no?
- 4 ¿Qué otros servicios de Azure probablemente integraría para que ejecutar sus aplicaciones?
- 5 Si le queda tiempo durante su almuerzo, pruebe uno de los aceleradores de soluciones Azure IoT en www.azureiotsolutions.com/Accelerators. Hay un escenario de simulación de dispositivos que crea una VM y sensores simulados, que es como el dispositivo Raspberry Pi simulado, pero mucho más grande. Tarda unos minutos en aprovisionar todos los recursos necesarios, pero luego observe Azure Portal para ver lo que se creó y cómo todas las partes trabajan juntas:
- 6 ¿Puede ver cómo se utilizan los servicios de los capítulos anteriores, como almacenamiento y Cosmos DB?
- 7 ¿Qué otros aceleradores de soluciones de IoT están disponibles? ¿Alguno de ellos se alinea con las ideas que tenía para sus propias aplicaciones?

Informática sin servidor

En este capítulo final, miraremos al futuro con la informática sin servidor. Si usted es un desarrollador, la idea de los contenedores (examinados en el capítulo 19) puede haber sido atractiva porque es menos necesario configurar la infraestructura subyacente para sus aplicaciones. Si es así, le van a encantar los componentes sin servidor de Azure. Y si usted es un administrador de TI que repentinamente se pregunta qué incluirá su trabajo si no hay servidores en el futuro, no se preocupe. La *informática sin servidor* puede ser más que un término de marketing, ya que muchas de las habilidades de servidor y de infraestructura que usted tiene continuarán aplicándose.

En Azure, dos ofertas principales proporcionan características de cálculo sin servidor: Azure Logic Apps y Azure Function Apps. En este capítulo, exploraremos lo que cada servicio ofrece y cómo pueden trabajar juntos. Para asegurarse de que sus aplicaciones sin servidor puedan comunicarse entre sí y distribuir datos, también analizaremos los servicios de mensajería como Azure Event Grid, Service Bus y Event Hubs.

21.1 *¿Qué es la informática sin servidor?*

Decir que la informática sin servidor no tiene servidor es erróneo: un servidor, en algún lugar, ejecuta un código para usted. La diferencia de las cargas de trabajo de aplicaciones de IaaS, como las VM de Azure y las cargas de trabajo de PaaS en aplicaciones web, es que las aplicaciones sin servidor suelen desglosarse en unidades discretas más pequeñas de una aplicación. No se ejecuta una sola aplicación grande; en su lugar, ejecuta componentes de la aplicación breves. Si esto suena similar a los contenedores y los microservicios que analizamos en el capítulo 19, no se preocupe que no se está volviendo loco: la informática sin servidor se superpone un montón con esos temas en términos de cómo diseñar sus aplicaciones. Podrás crear microservicios utilizando los enfoques sin servidor que veremos en este capítulo.

En la figura 21.1, se muestra cómo una aplicación se desglosa en pequeños componentes que se ejecutan en un proveedor informático sin servidor y proporcionan pequeñas unidades de salida.



Figura 21.1 En un entorno informático sin servidor, cada aplicación se desglosa en unidades pequeñas y discretas de componentes de la aplicación. Cada componente se ejecuta en un proveedor de informática sin servidor, como Azure Function Apps, y se produce la salida que puede ser consumida por otros componentes de la aplicación sin servidor u otros servicios Azure como Azure IoT o almacenamiento Azure.

En Azure, la informática sin servidor incluye dos servicios principales:

- *Azure Logic Apps*: para responder a ciertas entradas y desencadenantes, las aplicaciones lógicas le permiten crear flujos de trabajo que pueden procesar y generar acciones adicionales de apuntar y hacer clic, sin necesidad de código. Las aplicaciones lógicas pueden ser creadas por usuarios sin ninguna experiencia en programación ni infraestructura de TI. En la figura 21.2, se muestra un esquema simple de aplicación lógica.

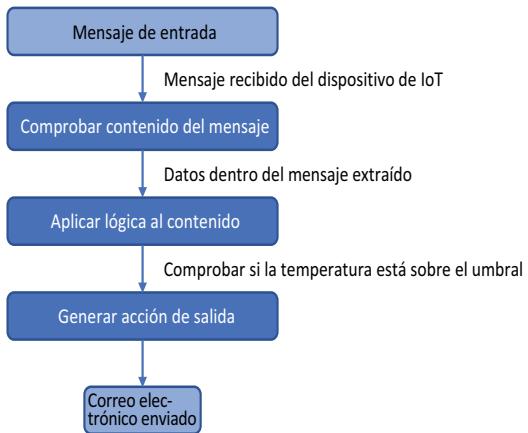


Figura 21.2 En una aplicación lógica, una entrada podría ser cuando se publica un tweet, se carga un archivo o se recibe un mensaje de un dispositivo de IoT. La aplicación lógica aplica reglas y filtros a los datos y determina si el mensaje cumple los criterios definidos por usted. Luego, se completan las acciones de salida, como la generación de un correo electrónico. Toda esta lógica no implica ninguna programación ni infraestructura de aplicación que no sea una suscripción a Azure.

No hay actualizaciones de seguridad que mantener y no hay requisitos de diseño en torno a la alta disponibilidad o la capacidad de escalar. La plataforma Azure se encarga automáticamente de esto. Existen cientos de conectores preconfigurados para aplicaciones lógicas que se integran con servicios como Twitter, Office 365, SharePoint y Outlook. Puede responder a los tweets públicos acerca de su empresa o producto, enviar un mensaje de alerta por correo electrónico cuando se cargue un archivo en SharePoint, o mandar una notificación cuando se reciba un mensaje de un dispositivo de IoT.

- *Azure Function Apps*: para ejecutar pequeños bloques de código, las aplicaciones de función le permiten utilizar lenguajes de programación comunes como C#, Node.js y Python sin ninguna administración de infraestructura adicional.

Su código se ejecuta en un entorno seguro y aislado, y se le factura basándose en el consumo de memoria por segundo. En la figura 21.3, se describe el proceso básico para una aplicación de función.

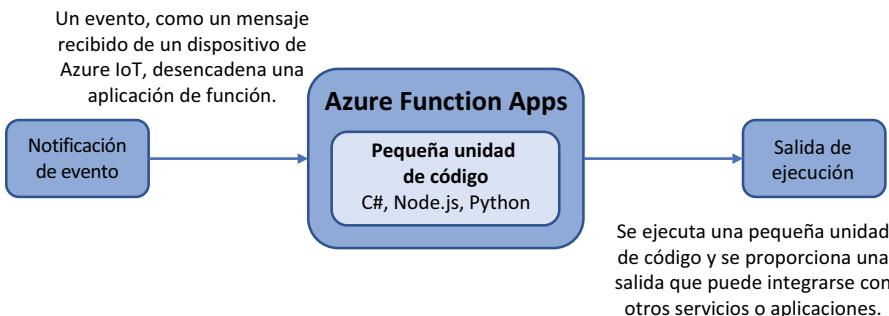


Figura 21.3 Al igual que con una aplicación lógica, una notificación de evento o desencadenante inician, en general, una función de Azure. La aplicación de función contiene una pequeña unidad de código que ejecuta una tarea específica. No hay ninguna infraestructura para configurar o mantener. Solo se necesita su pequeño bloque de código. Cuando finalice la ejecución del código, la salida se puede integrar con otro servicio o aplicación Azure.

No hay VM para mantener y no se necesita ninguna aplicación web. No tiene que preocuparse por la alta disponibilidad o la escala, porque el servicio Azure Function Apps se ocupa de estas tareas. Todo lo que proporciona es su código, y la plataforma Azure se asegura de que cada vez que necesite ejecutar ese código, los recursos estén disponibles para procesar su solicitud.

Las aplicaciones lógicas no requieren código, por lo que tienen una mayor base de usuarios potenciales. Los propietarios de aplicaciones de negocios o los equipos de finanzas y contabilidad, por ejemplo, pueden desarrollar sus propias aplicaciones lógicas sin tener que escribir código. Las aplicaciones de función proporcionan más control y flexibilidad y le permiten manipular eventos de una manera específica y mejorar la integración con otros componentes de la aplicación.

Tanto las aplicaciones lógicas como las aplicaciones de función proporcionan una manera para que usted realice acciones basadas en desencadenantes sin tener que mantener un entorno de aplicación o infraestructura. Un servidor en algún lugar en Azure ejecuta su aplicación lógica o de función, pero desde su perspectiva como administrador o desarrollador de TI, estas son tecnologías sin servidor.

21.2 Plataformas de mensajes Azure

En el capítulo 12, analizamos cómo controlar y solucionar problemas de los recursos de Azure, y en el capítulo 16 vimos cómo utilizar Azure Security Center para detectar problemas y realizar la administración de actualizaciones. Ambas características se basan en flujos de datos, como la extensión de diagnóstico de VM de Azure, para informar a la plataforma lo que está sucediendo en la VM. La plataforma de diagnóstico y control de Azure es genial, y otros servicios como Web Apps, Azure Container Instances y Azure IoT Hub también pueden transmitir diagnósticos de servicio para su análisis central.

Con aplicaciones sin servidor, a menudo necesita una manera de intercambiar mensajes y transmitir datos reales de la aplicación, no solo para solucionar problemas de diagnóstico o actualizaciones de estado. Ahí es cuando necesita una plataforma de mensajería.

21.2.1 Azure Event Grid

¿Qué pasa si solo desea informar sobre ciertas acciones o actividades que se están completando? En los flujos de trabajo de automatización y la informática sin servidor, la capacidad de realizar una acción en respuesta a un evento es útil, como se muestra en la figura 21.4.

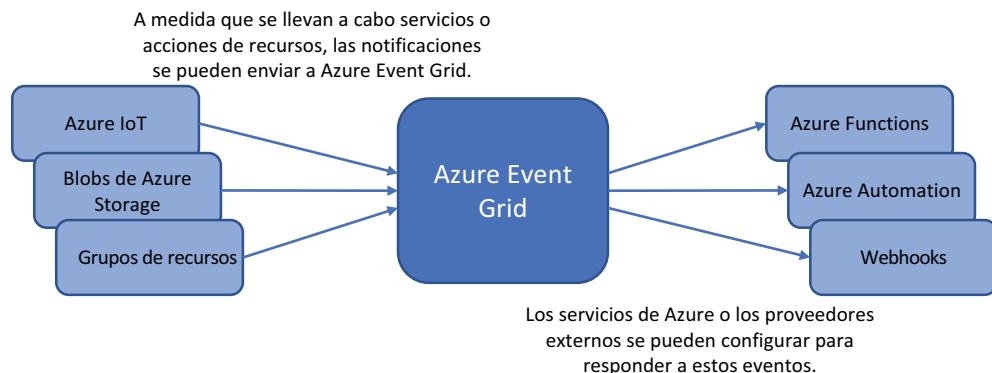


Figura 21.4 Los servicios Azure como Azure IoT y el almacenamiento Azure pueden enviar notificaciones a Azure Event Grid. Estas notificaciones pueden ocurrir cuando se recibe un mensaje de un dispositivo de IoT o se carga un archivo en el almacenamiento. Azure Event Grid permite a otros servicios y proveedores suscribirse a estas notificaciones para realizar acciones adicionales en respuesta a eventos.

Analicemos un par de escenarios que puede utilizar en su pizzería:

- *Mensaje recibido en un IoT Hub:* un dispositivo de IoT conectado IoT Hub puede informar una lectura de temperatura en un horno o en la ubicación de un vehículo de entrega. IoT Hub está configurado para reenviar una notificación a Azure Event Grid.

Una función de Azure se suscribe a las notificaciones de Event Grid para IoT Hub y ejecuta un pequeño componente de la aplicación sin servidor para registrar la información en Cosmos DB y enviar una notificación por correo electrónico. También puede utilizar Logic Apps en lugar de Azure Function Apps, según la complejidad que necesita la respuesta de la aplicación.

- *Archivo cargado en Azure Storage:* el departamento de marketing puede subir al almacenamiento un cupón promocional para ahorrar dinero en una orden de pizza. Cuando se crea un nuevo archivo, se envía una notificación a Event Grid.

Un webhook se suscribe a Event Grid y publica una copia de la imagen de almacenamiento en Twitter. Este tweet permite a los clientes conocer la oferta de la semana o el cupón de ahorro de dinero.

Estos escenarios son para escenarios informáticos sin servidor verdaderamente automático, pero Event Grid también puede integrarse con recursos más tradicionales como VM y aplicaciones web. Por ejemplo, se puede configurar un grupo de recursos para enviar notificaciones a Event Grid. Hay muchas maneras de crear una VM, como en el portal, con la CLI de Azure, o con una plantilla del Administrador de recursos, por lo que desea asegurarse de que la VM esté correctamente configurada para Update Management a través de Security Center. Se puede suscribir un runbook de automatización Azure a Event Grid para recibir notificaciones sobre las operaciones de creación de VM y, a continuación, incorporado en la VM, al servicio de administración de actualizaciones e instalar las actualizaciones de aplicaciones o seguridad necesarias.

21.2.2 Azure Event Hubs y Service Bus

Event Grid puede funcionar con muchos recursos Azure y es muy adecuado para la informática sin servidor con aplicaciones lógicas o aplicaciones de función. Pero las aplicaciones lógicas y las aplicaciones de función pueden ejecutarse basándose en otras entradas de datos, como hubs de eventos o un Service Bus. Veamos las diferencias entre estos diversos servicios de mensajería para que usted pueda decidir cuándo utilizarlos:

- *Azure Event Hubs* le permite recibir una secuencia de datos, como por ejemplo dispositivos de IoT o telemetría de aplicaciones. Los hubs de eventos proporcionan una plataforma de mensajería de baja latencia capaz de manipular millones de eventos por segundo de múltiples proveedores simultáneos. Los hubs de eventos son un almacén de datos más que una cola de mensajes, y el cliente o la aplicación comprueba los eventos en el hub con la frecuencia que usted deseé. Luego, los datos recibidos en el hub de eventos pueden ser procesados por otros servicios, como se muestra en la figura 21.5.
- *Azure Service Bus* permite a los componentes de la aplicación intercambiar datos de mensajes, como las colas de almacenamiento que examinamos en el capítulo 4. Las colas de almacenamiento de información son una implementación anterior y más básica de una plataforma de mensajería en Azure. Un *Service Bus*

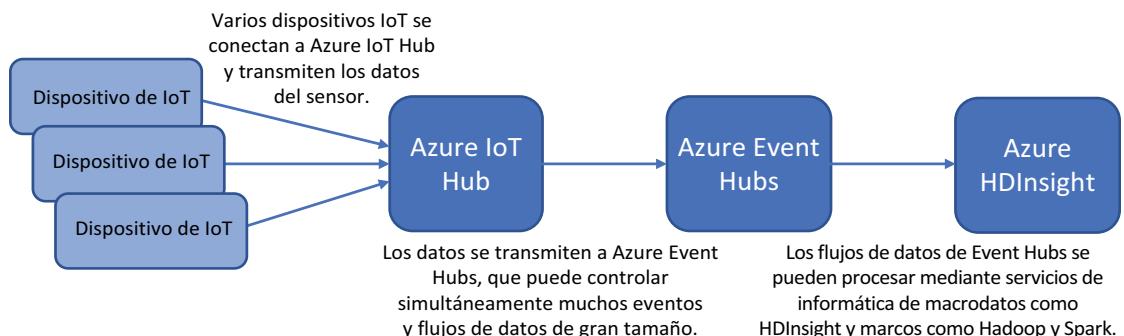


Figura 21.5 Los dispositivos de IoT se conectan al IoT Hub y pueden transmitir todos sus datos de sensores. Podría haber cientos o miles de dispositivos de IoT conectados. Azure Event Hubs manipula todos estos flujos de datos separados y permite que servicios como Azure HDInsight procese los datos sin procesar en clústeres de Hadoop o Spark para analizar y generar informes.

proporciona funciones más avanzadas, como pedidos garantizados de mensajes, operaciones atómicas y envío de mensajes en lotes. En la figura 21.6, se describe un escenario común para un Service Bus.

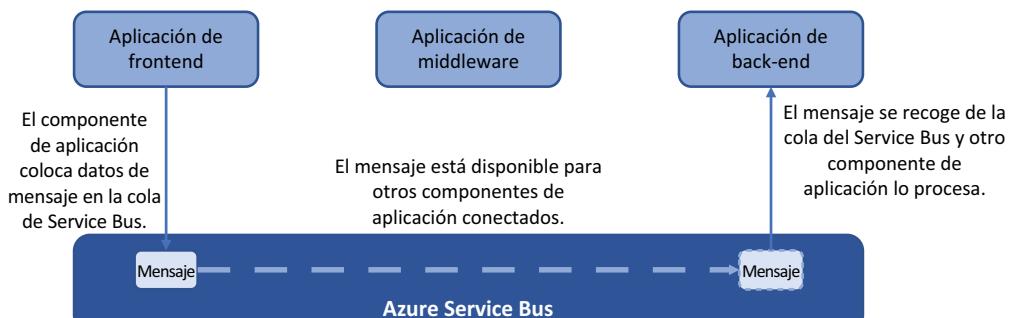


Figura 21.6 Los mensajes se colocan en una cola de Service Bus por componentes de aplicaciones, una aplicación frontend, en este ejemplo. Otras aplicaciones de middleware o back-end pueden recoger estos mensajes y procesarlos según sea necesario. Aquí, una aplicación back-end recoge el mensaje y lo procesa. Las funciones avanzadas de mensajería incluyen la garantía del orden de los mensajes en la cola, los mensajes de bloqueo, los tiempos de espera y los relés.

Con tres servicios que le permiten transmitir, recibir y procesar datos entre aplicaciones y servicios en Azure, ¿cuál se utiliza y cuándo? En la tabla 21.1, se proporciona una recapitulación de alto nivel de Event Grid, Event Hubs y Service Bus.

Tabla 21.1 Cada servicio está diseñado para cubrir un escenario diferente. Event Grid le permite reaccionar ante los eventos, los Event Hubs le permiten transmitir grandes cantidades de datos y Service Bus le permite enviar mensajes entre los servicios y los componentes de la aplicación.

Servicio de Azure	Proporciona	Caso de uso
Event Grid	Distribución de eventos	Realice una acción adicional basándose en la aparición de un evento.
Event Hubs	Flujos de datos	Reciba y transmita grandes volúmenes de datos simultáneos.
Service Bus	Transmisión de mensajes	Proporcione comunicación entre servicios y aplicaciones.

Azure Logic Apps y Function Apps pueden ser desencadenados por las tres plataformas de mensajería. Vamos a crear un service bus que se puede utilizar para desencadenar una aplicación lógica.

21.2.3 Creación de un Service Bus y su integración con un IoT Hub

En este escenario, utilicemos un Service Bus para transmitir mensajes recibidos de un IoT Hub. El dispositivo Raspberry Pi simulado del capítulo 20 genera lecturas de temperatura y las transmite al IoT Hub. Si la temperatura es superior a 30 °C, se incluye

otro dato en el mensaje del dispositivo de IoT: `temperatureAlert = true`. En la figura 21.7, se describe cómo se puede integrar un IoT Hub con el Service bus para procesar mensajes con esta alerta de temperatura.



Figura 21.7 Cuando el dispositivo de IoT Raspberry Pi simulado envía datos de mensajes, una lectura de temperatura de 30 °c o más genera una alerta. Los mensajes etiquetados con esta alerta se colocan en un Service Bus. Luego, estos mensajes se pueden utilizar para desencadenar aplicaciones lógicas.

Pruébelo ahora

Para crear un Service Bus, complete los pasos siguientes:

- 1 Abra Azure Portal y seleccione Crear un recurso en la parte superior izquierda del menú.
- 2 Busque y seleccione Service Bus y, a continuación, elija Crear.
- 3 Proporcione un nombre, como `azuremol`, y, a continuación, seleccione el nivel básico de precios.
- 4 Cree nuevo grupo de recursos y proporcione un nombre, como `azuremol-chapter21`. Asegúrese de que la ubicación sea la misma que para los recursos creados en el capítulo 20, como Este de EE. UU. La interacción entre una cola de Service Bus, una aplicación lógica y una aplicación de función puede tener problemas si no es coherente con sus ubicaciones.
- 5 Acepte los demás valores predeterminados y elija crear Service Bus.
- 6 Cuando cree el recurso, seleccione el grupo de recursos y, a continuación, elija el Service Bus que creó en el paso 5.
- 7 Seleccione Colas; agregue una nueva cola; y escriba un nombre, como `azuremol`.
- 8 Acepte todos los demás valores predeterminados y elija Crear.

Con un Service Bus y una cola creados, ¿cómo configurar un IoT Hub para utilizarlos? En el IoT Hub, define los *puntos de conexión* como los destinos de los mensajes recibidos de los dispositivos de IoT. Existe un punto de conexión predeterminado en el concentrador de IoT para todos los mensajes que no cumplen los criterios definidos. Puede configurar el Service Bus como punto de conexión para recibir mensajes. A continuación, se define una *ruta* que incluye criterios para los cuales los mensajes deben dirigirse a un punto de conexión. En este ejemplo, esta ruta

requiere cualquier mensaje que contenga `temperatureAlert = true` en el cuerpo del mensaje que se enrutará al punto de conexión de Service Bus, como se muestra en la figura 21.8.

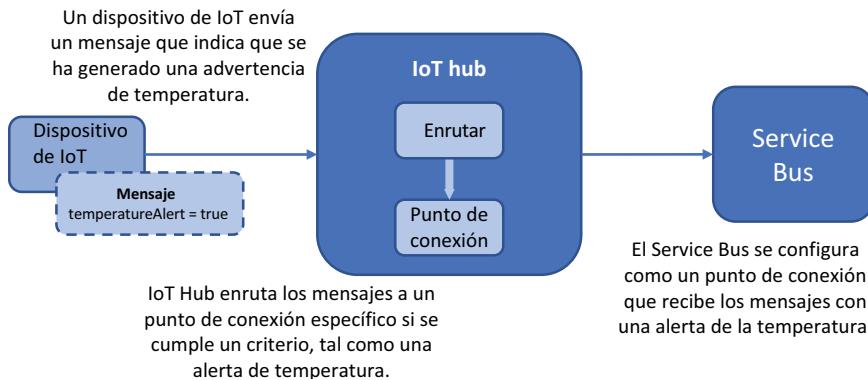


Figura 21.8 A medida que los mensajes se transmiten desde dispositivos de IoT a un IoT Hub, se pueden enrutar a puntos de conexión específicos basándose en los criterios que defina. Los mensajes que contienen una alerta de temperatura en el cuerpo del mensaje se pueden enrutar a un punto de conexión que utilice la cola de Service Bus. Luego, los mensajes colocados en la cola de Service Bus que contienen una alerta de temperatura se pueden utilizar para desencadenar cosas como Azure Logic Apps o Azure Function Apps.

Pruébelo ahora

Para configurar un IoT Hub para enrutar mensajes de alerta de temperatura al Service Bus, complete los pasos siguientes:

- 1 Seleccione el grupo de recursos del capítulo 20, como `azuremolchapter20` y, a continuación, elija el IoT Hub.
 - 2 En mensajería en la barra de navegación a la izquierda, seleccione Enrutamiento de mensajes, y elija para agregar un punto de conexión personalizado para una cola de Service Bus.
 - 3 Proporcione un nombre de punto de conexión, como `azuremol`.
 - 4 Seleccione el espacio de nombres de la cola de Service Bus, como `azuremol` y, a continuación, la cola real.
 - 5 Para dirigir mensajes a este punto de conexión, cree una ruta. En la sección Enrutamiento de mensajes en la barra de navegación a la izquierda, seleccione Rutas y elija para agregar una nueva ruta.
 - 6 Proporcione un nombre, como `temperatureAlert`.
 - 7 Elija el punto de conexión de Service Bus que creó en el paso anterior, como `azuremol`.
 - 8 Para la consulta de enrutamiento, escriba lo siguiente:
- ```
temperatureAlert = "true"
```
- 9 Cuando esté listo, guarde la ruta.

Ahora tiene un dispositivo Raspberry Pi simulado que envía datos al IoT Hub, así como una ruta para colocar mensajes que contengan una alerta de temperatura en una cola de mensajes del Service Bus. En realidad no tiene una aplicación aún: no hay nada que pueda hacer con los datos de la cola de Service Bus. ¿Qué podría querer hacer con una alerta de temperatura? El envío de una notificación por correo electrónico es un ejemplo común, así que vamos a ver cómo se puede desencadenar una aplicación lógica cada vez que se coloca un mensaje en la cola de Service Bus.

### 21.3 Creación de una aplicación lógica de Azure

Como vimos cuando analizamos aplicaciones lógicas en la sección 21.1, un mensaje recibido de una cola de Service Bus puede utilizarse como un desencadenador para iniciar el proceso de ejecución. Se utiliza el IoT Hub para procesar los mensajes recibidos de los dispositivos de IoT y solo enrutar a los mensajes del punto de conexión de la cola de Service Bus que contienen `temperatureAlert = true` en el cuerpo del mensaje. Con este enfoque, la aplicación lógica solo se ejecuta cuando se genera una alerta de temperatura.

En la figura 21.9, describe lo que hace su aplicación lógica. Cuando se coloca un mensaje en la cola de Service Bus, la aplicación lógica se ejecuta y envía una alerta por correo electrónico.

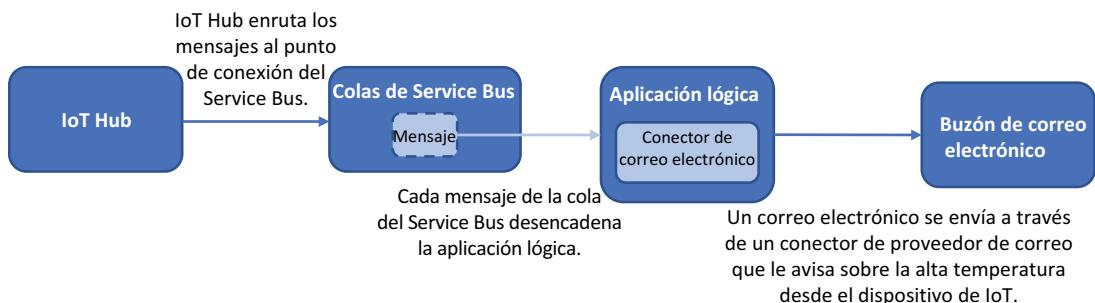


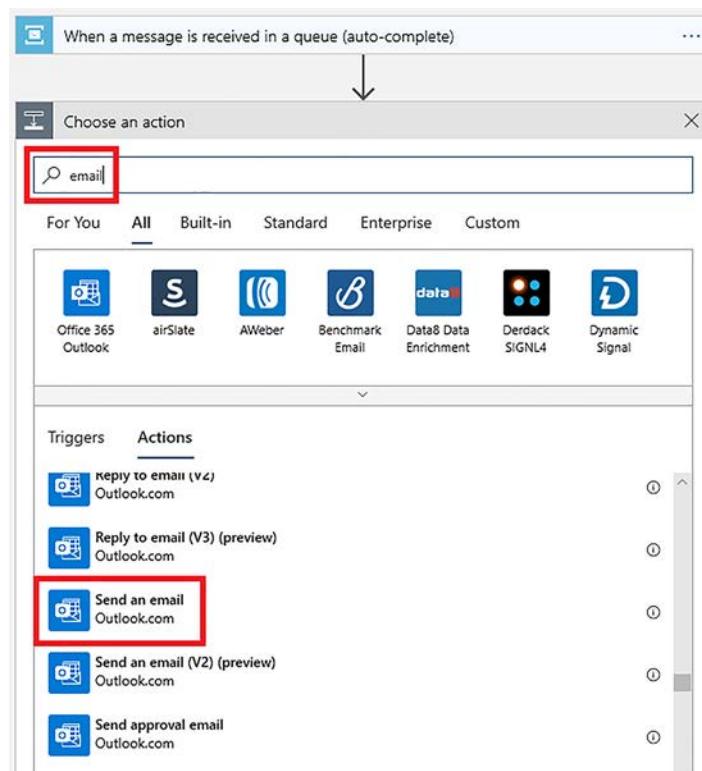
Figura 21.9 Cada mensaje recibido en la cola del Service Bus desde el IoT Hub desencadena la aplicación lógica. Cuando se ejecuta la aplicación lógica, envía una notificación por correo electrónico a través de un proveedor de correo definido.

#### Pruébelo ahora

Para crear una aplicación lógica, complete los pasos siguientes:

- 1 Abra Azure Portal y elija Crear un recurso en la parte superior izquierda del menú.
- 2 Busque y seleccione Logic App y, a continuación, elija Crear.
- 3 Proporcione un nombre, como `azuremol`, y seleccione el grupo de recursos, como `azuremolchapter21`. Una vez más, elija la misma ubicación que sus otros recursos de IoT del capítulo 20.
- 4 Acepte los otros valores predeterminados y elija Crear.
- 5 Cuando haya creado el recurso, seleccione el grupo de recursos y, a continuación, abra la aplicación lógica. Para la opción "Agregar desencadenadores comunes", elija "Cuando se reciba un mensaje en una cola de Service Bus".

- 6 Proporcione un nombre, como azuremol; luego seleccione su cola de Service Bus, como azuremol.
- 7 Elija la directiva de Service Bus predeterminada que se indica, como RootManageSharedAccess-Key, y cree la conexión.
- 8 Seleccione Continuar y, a continuación, elija el nombre de la cola del Service Bus, como azuremol.
- 9 Acepte los valores predeterminados, como la frecuencia para comprobar si hay mensajes.
- 10 Elija para agregar un nuevo paso a la aplicación lógica.
- 11 Para agregar una acción, busque lo que desea hacer. En este ejercicio, busque *correo electrónico*. Seleccione su proveedor, como Gmail: enviar un correo electrónico, Outlook.com: enviar un correo electrónico, o SMTP: enviar un correo electrónico, como se muestra en la figura 21.10.



**Figura 21.10** Busque y seleccione su proveedor de correo electrónico actual, como Outlook.com o Gmail. También puede elegir SMTP: enviar un correo electrónico para configurar manualmente un proveedor diferente.

- 12 Inicie sesión en su proveedor de correo electrónico para autorizar el enruteamiento del correo y confirme que desea conceder permisos de aplicaciones lógicas para enviar correo electrónico.

- 13 Proporcione una dirección de correo electrónico de destinatario en la que reciba correo electrónico; un asunto de correo electrónico, como Alerta de temperatura y un cuerpo del mensaje, como Temperatura alta detectada en el dispositivo de IoT.
- 14 Guarde la aplicación lógica.

Hagamos una pausa para revisar lo que ha desarrollado en los últimos ejercicios, como se muestra en la figura 21.11. Este diseño básico de aplicaciones sin servidor no incluye ningún control que limite la cantidad de mensajes que se enviarán. En la aplicación lógica, puede definir que solo desea enviar un máximo de cinco alertas por correo electrónico y esperar 30 minutos antes de enviar más. Como parte de su diseño de la aplicación, usted debe considerar cómo desea ser notificado de situaciones como esta. También puede configurar la aplicación lógica para leer los datos del mensaje de la cola del Service Bus e incluir la marca de tiempo del mensaje del dispositivo de IoT y la temperatura real registrada. Analizaremos cómo hacer esto en el siguiente ejercicio.

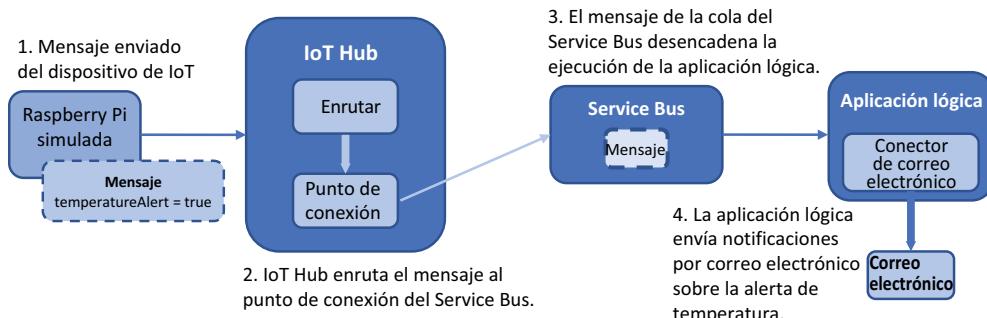


Figura 21.11 El dispositivo Raspberry Pi simulado envía un mensaje al IoT Hub cada dos segundos que contiene lecturas de sensores de temperatura. Si la temperatura es superior a 30 °C, se indica una alerta de temperatura. El IoT Hub enruta cualquier mensaje que contenga una alerta de temperatura a una cola de Service Bus. Los mensajes en esta cola desencadenan una aplicación lógica de Azure para ejecutar. La aplicación lógica está conectada a un proveedor de correo electrónico, como Outlook o Gmail, y envía una notificación por correo electrónico acerca de la advertencia de temperatura del dispositivo de IoT.

Veamos esta aplicación básica sin servidor en acción.

### Pruébelo ahora

Para ejecutar el dispositivo Raspberry Pi simulado y probar su aplicación lógica, complete los pasos siguientes.

- 1 Abra un navegador web al dispositivo Raspberry Pi IoT simulado del capítulo 20 (<https://azure-samples.github.io/raspberry-pi-web-simulator>).
- 2 Compruebe que la cadena de conexión de su IoT Hub se sigue agregando en la ventana de código que configuró en el capítulo 20.

**3** Elija para ejecutar la aplicación.

Las lecturas simuladas del sensor de temperatura y humedad se generan cada dos segundos, y se envía un mensaje al IoT Hub. Puede tomar algunos mensajes antes de que se genere una lectura de temperatura simulada de 30 °C y se muestre en la ventana de salida.

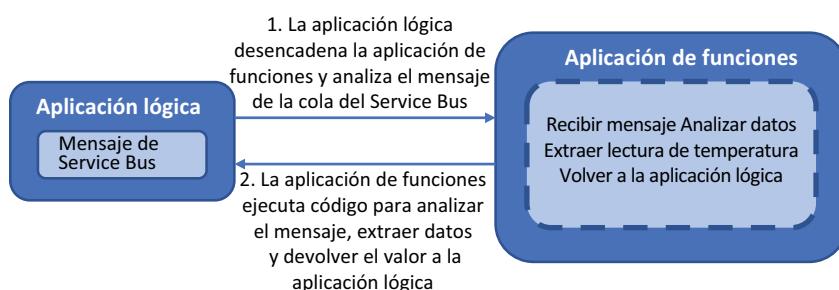
El IoT Hub enruta cualquier mensaje que contenga `temperatureAlert: true` al punto de conexión de Service Bus. A medida que estos mensajes se colocan en la cola de Service Bus, la aplicación lógica los recoge y envía un correo electrónico a través del proveedor definido. A continuación, recibe un correo electrónico para notificarle de una lectura de alta temperatura. Este proceso solo debe tomar unos segundos en completarse.

**4** El dispositivo Raspberry Pi simulado genera mensajes cada dos segundos, de modo que detenga la aplicación a menos que le gusten un montón de alertas por correo electrónico.

Cuando recibe alertas por correo electrónico, el mensaje no contiene mucha información. Su aplicación lógica no extrae el contenido del mensaje del Service Bus y formatea la información. Sería genial si el correo electrónico de alerta pudiese incluir el nombre del dispositivo de IoT o la temperatura registrada. ¿Cómo se puede procesar cada mensaje y realizar un análisis sobre este? ¿Qué pasa con el otro servicio sin servidor de Azure que analizamos: Azure Function Apps?

## 21.4 Creación de una Azure Function App para analizar datos del dispositivo de IoT

Para ampliar su aplicación sin servidor actual, puede desencadenar una aplicación de función de Azure desde dentro de su aplicación lógica. Los datos del mensaje del Service Bus se pueden enviar a una aplicación de función para analizar la temperatura registrada. Luego, la notificación por correo electrónico enviada por la aplicación lógica puede incluir información sobre el nombre del dispositivo de IoT y la temperatura registrada. La interacción entre la aplicación lógica y la aplicación de función se muestra en la figura 21.12.



**Figura 21.12** La aplicación lógica desencadena la aplicación de funciones. El mensaje recibido en la cola del Service Bus se pasa a la función. El código de la aplicación de función analiza el mensaje, extrae la temperatura y devuelve ese valor a la aplicación lógica. Demora unos pocos milisegundos para que la aplicación de funciones ejecute este código, por lo que el costo de realizar estas tareas de cálculo es una fracción de un centavo.

### Pruébelo ahora

Para crear una aplicación de funciones y activarla desde la aplicación lógica, complete los pasos siguientes:

- 1 Abra Azure Portal y seleccione Crear un recurso en la parte superior izquierda del menú.
- 2 Busque y seleccione Function App y, a continuación, elija Crear.
- 3 Seleccione un grupo de recursos, como azuremolchapter21 y proporcione un nombre, como azuremol. Quiere estar en la misma región que sus recursos anteriores.

Quiere publicar código, considere que también puede publicar una imagen de contenedor Docker (capítulo 19). Ni siquiera necesitaría crear una instancia de contenedor ni ninguna infraestructura adicional; un contenedor de corta duración se ejecutaría según sea necesario y luego se detendrá.

- 4 Para esta aplicación básica, elija el tiempo de ejecución Node.js, ya que utilizamos algo de JavaScript simple.
- 5 Tiene tres opciones de planes de hospedaje. Un *plan de consumo* le permite pagar por ejecución y los recursos que necesita se asignan dinámicamente en tiempo de ejecución. Para aplicaciones más coherentes y listas para la producción, puede utilizar un *plan de hospedaje dedicado* o *premium* que proporcione un costo más fijo y predecible. Los planes premium proporcionan características adicionales, como asegurar la conectividad a un conjunto definido de redes virtuales de Azure y tener siempre una instancia lista para evitar algunos retrasos en un escenario de arranque en frío para su aplicación. Para este ejercicio, elija un plan de consumo.
- 6 Acepte los demás valores predeterminados para crear una cuenta de almacenamiento con nombre y Application Insights y, a continuación, elija Revisar + Crear.
- 7 Cuando esté listo, cree la aplicación de función. Tarda un minuto o dos para crear la aplicación de función.
- 8 Cuando haya creado el recurso, seleccione el grupo de recursos y, a continuación, abra la aplicación lógica del ejercicio anterior y seleccione Editar.
- 9 En el diseñador de Logic Apps, elija para agregar un nuevo paso.
- 10 Busque y seleccione Azure Functions y, a continuación, elija la función creada en los pasos anteriores (como azuremol) y elija Crear nueva función.
- 11 Proporcione un nombre de función, como analyzeTemperature.
- 12 Elimine cualquier código existente, reemplácelo con el código de los siguientes listados y, a continuación, elija Crear.

Este código también está disponible en el repositorio de GitHub en [w](#).

### Listado 21.1 Código analyzeTemperature de JavaScript para una aplicación de funciones

Crea la función. Cada aplicación de funciones JavaScript comienza con la exportación de una función que contiene un objeto de contexto. Este objeto de contexto se utiliza para pasar datos de un lado a otro.

```
module.exports = function (context, req) {
 var buffer = new Buffer(req.body.ContentData, 'base64');
 var decodedString = buffer.toString();
 var objects = JSON.parse(decodedString);
 var temperature = objects["temperature"];
 context.res = {
 body: {
 analysis: "Recorded temperature was " + temperature + "!"
 }
 };
 context.log("Recorded temperature was " + temperature);
 context.done();
};
```

Lee el contenido del mensaje desde el Service Bus.

Crea un objeto JSON de mensajes de Service Bus descifrados

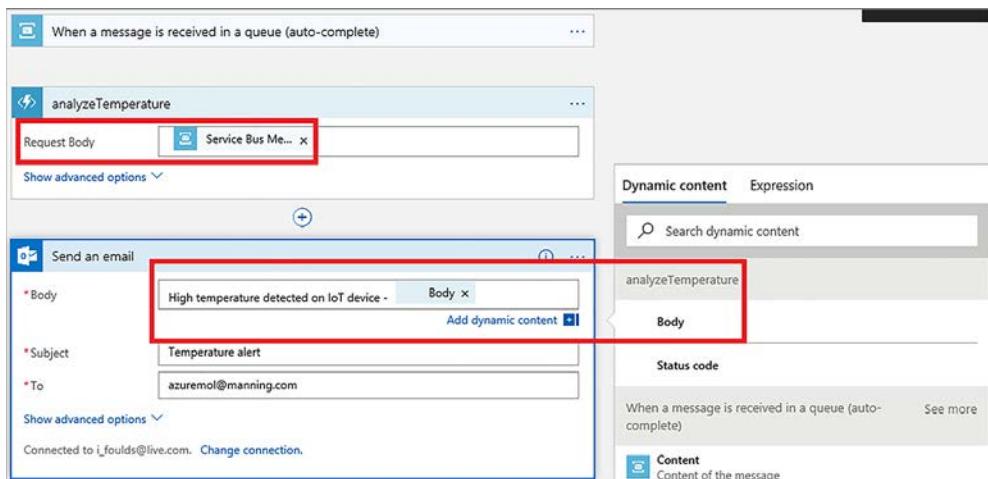
Extrae la temperatura registrada del dispositivo de IoT

Genera una respuesta para enviar de vuelta a la aplicación lógica

Finaliza la función. Cada aplicación de funciones JavaScript debe terminar con una llamada a context.done(), que indica a la aplicación de funciones que el código está terminado.

Descifra desde Base64  
Envía la temperatura al registro de la consola

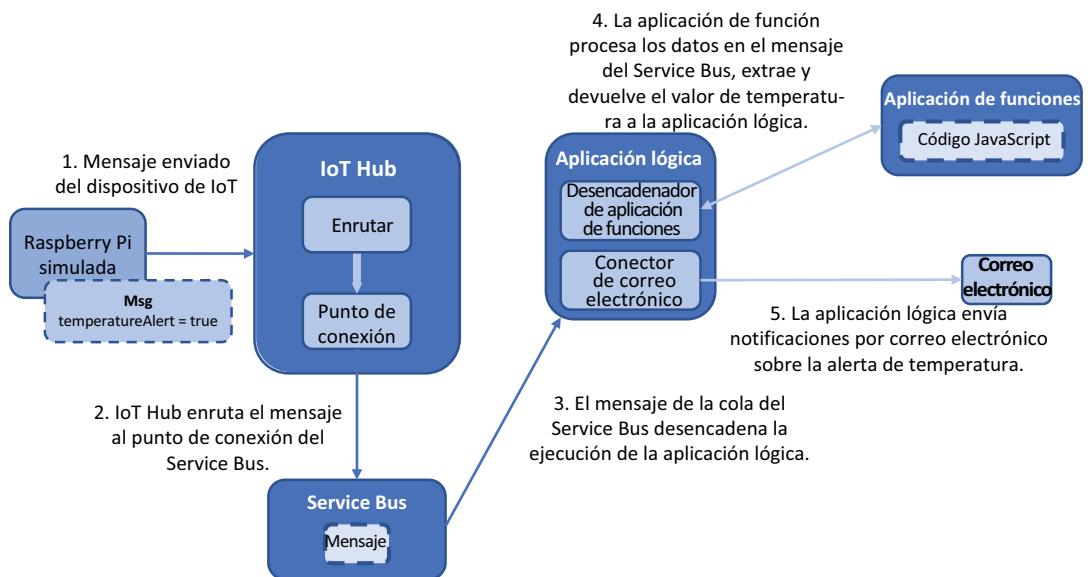
- 13 De nuevo en el diseñador de Logic Apps para el paso de la función, seleccione el cuadro Solicitar cuerpo y elija Mensaje de Service Bus en la lista de contenido dinámico en el lado derecho.
- 14 En el diseñador de Logic Apps, arrastre y suelte para reordenar los pasos de modo que la acción Enviar un correo electrónico esté por debajo del paso de la aplicación de funciones analyzeTemperature, como se muestra en la figura 21.13.
- 15 Seleccione la acción Enviar un correo electrónico y, a continuación, seleccione el cuadro de texto para el cuerpo del mensaje de correo electrónico.



**Figura 21.13** Arrastre la acción Enviar un correo electrónico debajo de la función analyzeTemperature. Seleccione el final del Cuerpo del mensaje y aparecerá el cuadro de diálogo de contenido dinámico. Para insertar el valor de temperatura calculado por la aplicación de funciones, seleccione el Cuerpo del mensaje de la función analyzeTemperature.

- 16 En la función `analyzeTemperature`, seleccione la respuesta del Cuerpo, como se muestra en la figura 21.13.
- 17 En el diseñador de Logic Apps, elija Guardar.

Su aplicación sin servidor tiene muchas partes móviles. Examinemos lo que desarrolló antes de ejecutar el dispositivo de IoT Raspberry Pi simulado para generar alertas de correo electrónico que incluyan la lectura de la temperatura según lo calcula la aplicación de funciones. En la figura 21.14, se proporciona información general de todos los componentes que ahora se utilizan en la aplicación sin servidor.



**Figura 21.14** Como los mensajes se reciben desde el dispositivo Raspberry Pi simulado, cualquier mensaje que contenga una alerta de temperatura se enruta al punto de conexión de cola de Service Bus. Los mensajes en la cola de Service Bus desencadenan una aplicación lógica, que pasa el mensaje a una aplicación de funciones. Una función de JavaScript analiza la lectura de la temperatura y la devuelve a la aplicación lógica, que luego envía una notificación por correo electrónico que incluye la temperatura registrada por un sensor en el dispositivo de IoT.

- 18 Abra el dispositivo Raspberry Pi simulado en un navegador web y ejecute la aplicación. Cada vez que se genera la alerta de temperatura, la aplicación lógica desencadena la aplicación de funciones para extraer los datos de temperatura del cuerpo del mensaje e incluirlos en la notificación de correo electrónico. Puede tardar unos momentos para que una lectura de temperatura sea superior a 30 °C que luego marca el mensaje con una alerta de temperatura. Cuando se envía esa alerta y se procesa el mensaje, se recibe una notificación por correo electrónico que informa sobre la temperatura.

Respire profundo y dese una palmada en la espalda. ¡Eso fue mucho para hacer en su almuerzo!

### Errores de autenticación de su aplicación lógica a la aplicación funciones

Puede ver el historial de ejecución en la ventana de información general de su aplicación lógica en Azure Portal. Si recibe muchos errores repetidos, seleccione uno de los errores para ver más sobre donde se produce el error.

Un problema común es que la aplicación lógica no está autorizada automáticamente a hablar con la aplicación funciones. La reimplementación de la aplicación lógica suele corregir este error, pero la verdadera solución es probablemente agregar lo que se denomina clave de función al encabezado de la aplicación lógica.

Para obtener esta clave, seleccione su aplicación de funciones y, a continuación, elija la función que creó, como `analyzeTemperature`. En la opción Administrar, la tecla de función predeterminada puede visualizarse y copiarse. Copie esta clave, regrese a la aplicación lógica y abra el diseñador.

En la función `analyzeTemperature`, elija agregar un parámetro y, a continuación, agregue un encabezado. Desea enviar un poco de información al principio de la llamada a la aplicación de funciones que envía la clave. El proceso es un poco al revés, ya que introduce en par de claves, pero introduce `x-functions-key` para la clave, y luego pega su clave de función real como el valor.

La actualización de la integración de la aplicación lógica con la aplicación funcional tarda unos instantes. Después de eso, el historial de ejecución de la aplicación lógica debe mostrar los eventos que funcionan correctamente, y las notificaciones de correo electrónico deben iniciarse.

## 21.5 No deje de aprender

Este capítulo contiene un montón de nuevos conceptos. ¡De hecho, los capítulos anteriores contienen muchas nuevas ideas y tecnologías! No se preocupe si está luchando para entender cómo puede empezar a implementar todos estos servicios Azure, tales como contenedores, IA y ML, y la informática sin servidor. Estos capítulos se diseñaron para mostrar lo que es posible en Azure y demostrar que no es necesario limitarse a realizar una migración lift and shift de las aplicaciones heredadas. Al empezar a desarrollar y ejecutar aplicaciones en Azure, aproveche la oportunidad de modernizar las aplicaciones y revisar los flujos de trabajo de administración o implementación. Muchos servicios Azure que simplifican y aceleran el ciclo de vida de la aplicación; entonces, no sienta como si tuviera que quedarse con la ejecución de VM porque eso es lo que resulta cómodo en los negocios.

Sí, Azure ofrece muchos servicios nuevos y brillantes, pero en gran parte se basan en componentes básicos de infraestructura analizados en la parte 1 de este libro. Los desarrolladores pueden empezar a utilizar los últimos enfoques de diseño de aplicaciones que implican Kubernetes o informática sin servidor, y los administradores pueden reutilizar su conocimiento del centro de datos local con conceptos básicos de informática en la nube y técnicas de solución de problemas. A medida que crecen sus necesidades de negocio, Azure puede respaldarlas.

En el capítulo 1, fui abierto y sincero y dije que no cubriría todos los servicios en Azure. Hay muchos más servicios Azure para conocer y más para profundizar acerca de los servicios que vimos en el libro. Espero que haya encontrado al menos algunas áreas que le interesen y lo motiven a explorar un poco más. Mis favoritas incluyen conjuntos de escala de máquinas virtuales, Cosmos DB y Azure Kubernetes Service.

### 21.5.1 Materiales de aprendizaje adicionales

Soy parcial, pero creo que un buen lugar para continuar aprendiendo acerca de Azure es <https://docs.microsoft.com/azure>. Esta página web tiene la documentación del servicio principal de Azure, las guías de arquitectura, los recursos de referencia y SDK y los ejemplos. Cada servicio Azure tiene su propio conjunto de inicios rápidos, tutoriales y ejemplos, junto con información conceptual y guías individuales.

Si se pone serio, puede investigar las opciones de certificación para Azure. Los exámenes individuales incluyen *Microsoft Azure Administrator* (AZ-104), *Microsoft Azure Architect Technologies and Design* (AZ-303 y AZ-304), y *Microsoft Azure Security Technologies* (AZ-500). Este libro y los ejercicios de laboratorio que completó cubren muchas de las áreas en las que esos exámenes ponen a prueba sus conocimientos, pero tendría que estudiar algunas áreas adicionales de Azure AD y los procedimientos recomendados de diseño antes de hacer los exámenes. El sitio de Microsoft Learn en <https://docs.microsoft.com/learn> proporciona algunas vías de aprendizaje adicionales para las diferentes opciones de certificación de Azure para ayudarle a prepararse.

### 21.5.2 Recursos de GitHub

En este libro, utilizó ejemplos de código, plantillas y aplicaciones de ejemplo de <https://github.com/fouldsy/azure-mol-samples-2nd-edition>. Estas muestras deben mantenerse actualizadas a medida que se lancen las nuevas versiones de CLI de Azure; el repositorio de GitHub también incluye ejemplos y plantillas de PowerShell para todos los ejercicios. Este libro se centra en la CLI de Azure en Azure Cloud Shell, pero siéntase libre de explorar cómo se ve cada ejercicio en PowerShell o una plantilla.

Si nota algún problema con los ejemplos, cree un problema en GitHub en <https://github.com/fouldsy/azure-mol-samples-2nd-edition/issues>. Todo va muy rápido en Azure, y quiero asegurarme de que siempre tenga los últimos ejemplos de trabajo para ayudarlo a aprender. ¡Siéntase libre de hacer sugerencias, también! Todos los documentos Azure en <https://docs.microsoft.com/azure> también aceptan comentarios, problemas y ediciones, así que a medida que explore el resto de lo que ofrece Azure, siéntase libre de participar y ayudar a otros a aprender y crecer.

### 21.5.3 Reflexión final

Respire profundo y dese cuenta de que el cambio es normal. Se lanzan nuevas características y servicios casi diariamente. Azure, al igual que todos los principales proveedores de informática en la nube, puede verse y sentirse un poco diferente de la última vez que lo usó (hace una hora). Si tiene las habilidades fundamentales y la comprensión que espero que haya aprendido en este libro, puede adaptarse y crecer con todas las nuevas oportunidades que Azure ofrece. Siempre tiene algo nuevo que aprender, y me encantaría escuchar lo que desarrolla y ejecuta en Azure.



# índice

## Símbolos

carácter && 27  
objeto \$ResourceGroups 276  
objeto \$servicePrincipalConnection 275

## A

acceso interactivo de la consola de arranque 176  
ACI (Azure Container Instance) 284, 290, 292  
ACR (Azure Container Registry) 293  
activos, en Azure Automation 272–274  
actualizaciones 234–249  
    Azure Update Management 241–249  
        OMS (Operations Management Suite) 243  
        revisión y aplicación de actualizaciones 245–249  
    JIT (just-in-time) 237–241, 249  
        NSG de Azure Security Center 234  
actualizaciones JIT (Just-In-Time) 237–249  
acuerdos de nivel de servicio (SLA) 247  
Administrador de configuración local (LCM) 278  
agentes 180  
agrupación de recursos 80–81  
AKS (Azure Kubernetes Service) 284, 293–297  
    creación de clústeres con 294–295  
    ejecución de sitios web en Kubernetes 295–297  
    visualización de información en 294  
alertas 178–182  
alertas de métricas 182  
almacenamiento  
    almacenamiento de colas 55–56  
    disponibilidad de 56–57  
    en Azure 18  
    en VM 47–50  
        almacenamiento estándar versus almacenamiento premium 48–49  
    discos de datos 49–50

discos temporales 49–50  
opciones de almacenamiento en caché de disco 50  
redundancia 56–57  
almacenamiento (vDisk) 11  
almacenamiento con redundancia geográfica (GRS) 56  
almacenamiento con redundancia geográfica con acceso de lectura (RA-GRS) 57  
Almacenamiento de archivos 53  
Almacenamiento de blobs 52  
Almacenamiento de colas 53, 55–56  
almacenamiento de plantillas 87  
Almacenamiento de tablas 52 etiquetas administración de recursos con 80–81  
agrupación de recursos con 80–81  
almacenamiento en caché de lectura/escritura 50  
almacenamiento redundante a nivel local (LRS) 56  
almacenes de software 217–218  
almacenes, clave 219–221  
Amazon Web Services (AWS) 191  
analizador jq 226  
API (interfaces de programación de aplicaciones) 31  
API de REST 161  
aplicaciones  
    ciclos de vida de 76–77  
    Function Apps 328–331  
    Logic Apps 325–328  
    planes de servicio 38  
aplicaciones de equilibrio de carga 106–123  
aplicaciones escalables 124–140  
    escalamiento de aplicaciones web 136–139  
    beneficios de 124–129  
        escalamiento horizontal de recursos 128–129  
        escalamiento vertical de aplicaciones web 127–128  
        escalamiento vertical de las VM 125–127  
    conjuntos de escalamiento de VM 129–136  
        creación 131–133

- creación de reglas de escalamiento automático 133–136  
aplicaciones lógicas 35, 182, 325–328  
aplicaciones monolíticas 288  
aplicaciones. Consulte también Azure Web Apps  
Application Gateway 107–108  
Aprenda Docker en un mes de almuerzos (Stoneman) 297  
Aprenda Git en un mes de almuerzos (Umalí) 37  
APT (herramienta avanzada de empaquetado) 27  
Archivo de formato de objeto administrado (MOF) 280  
Archivo de MOF (formato de objeto administrado) 280  
Áreas de trabajo de Log Analytics 243  
asignación dinámica 62  
asignación estática 63  
Automation Hybrid Worker 272  
AWS (Amazon Web Services) 191  
Azure AD (Azure Active Directory) 222  
Azure Application Gateway 108  
Azure Application Insights 180  
Azure Automation 243, 269–283  
    activos 272–274  
    creación de cuentas en 271–272  
    descripción general de 179, 269–274  
PowerShell DSC 278–282  
    definición 280–282  
    Servidores de extracción de Azure Automation y 280–282  
runbooks 272–274  
    ejecución 276–277  
    muestra de 274–277  
    visualización de resultados de 276–277  
Azure Backup 191–201, 243  
    directivas y retención 193–196  
        RPO (objetivo de punto de recuperación) 194–195  
        RTO (objetivo de tiempo de recuperación) 195–196  
Programaciones de copia de seguridad 196–198  
restauración de VM 198–201  
    restauración completa de VM 199–201  
    restauración en el nivel de archivo 199  
Azure Bastion 24, 115  
Azure CLI 7, 12–13, 28, 31, 81–82, 152, 161  
Azure Cloud Shell 12–13  
Azure Cognitive Services 259–260  
Azure Container Instance. Consulte ACI  
Azure Container Registry. Consulte ACR  
Azure DNS (servicio de nombre de dominio) 158–162  
Azure Event Grid 320–321  
Azure Event Hubs 321–322  
Azure Firewall 238  
Azure Front Door 163–164  
Azure Function Apps 318  
Azure IoT (Internet de las Cosas) 300–316  
    creación de aplicaciones de función para analizar los datos de  
    descripción general de 300–302  
dispositivos 328–331  
Hub, administración centralizada de dispositivos con 303–309  
    integración con Service Bus 322–325  
    revisión de componentes 315  
    transmisión de datos de hub a aplicaciones web 309–315  
Azure IoT Edge 304–305  
Azure Key Vault 216–233, 304  
    almacenamiento de claves de cifrado en 211–213  
    creación de certificados 229–232  
    descripción general de 211  
    inyección de certificados 229–232  
    MSI (identidades de servicio administrado) 221–229  
    protección de la información en nubes 216–221  
    almacenes de software y HSM 217–218  
    creación de almacenes de claves y secretos 219–221  
Azure Kubernetes Service. Consulte AKS  
Azure Logic Apps 318  
Azure Monitor 243  
Azure Network Watcher 182–188  
    captura de paquetes de red 186–188  
    verificación de flujos de IP 183–184  
    visualización de reglas efectivas de NSG 184–186  
Azure Portal 12  
Azure PowerShell 11, 13, 31, 81–82, 161  
Azure Resource Manager 75–89  
    enfoque hacia 75–81  
        administración y agrupación de recursos con etiquetas 80–81  
        diseño alrededor del ciclo de vida de las aplicaciones 76–77  
        protección de recursos con bloques 79–80  
        protección y control de recursos 78–79  
    plantillas para 81–87  
        almacenamiento 87  
        creación 82–84  
        creación de múltiples tipos de recursos 84–85  
        herramientas para crear 85–86  
Azure Security Center 234, 249  
Azure Service Bus 310, 321–322  
Azure Service Fabric 289  
Azure Site Recovery 201–204, 243  
Azure Storage 47–57  
    agregar discos a VM 50–52  
    almacenamiento de VM 47–50  
        almacenamiento estándar versus almacenamiento premium 48–49  
        discos de datos 49–50  
        discos temporales 49–50  
        opciones de almacenamiento en caché de disco 50  
    beneficios de 52–57  
        almacenamiento de colas 55–56  
        almacenamiento de tablas 53–54  
        disponibilidad de almacenamiento 56–57  
        redundancia 56–57  
Azure Traffic Manager. Consulte Traffic Manager

- Azure Update Management 241–249  
  OMS (Operations Management Suite) 243  
  revisión y aplicación de actualizaciones 245–249
- Azure Web Apps 33–45  
  escalamiento 127–139  
  administración 42–44  
  compilación con tráfico seguro 68–72  
  creación de conexiones de red de acceso remoto 68–69  
  creación de VM 69–70  
  uso de agentes de SSH para conectarse a VM 70–72
- creación 37–42  
  creación de aplicaciones web básicas 37  
  implementación de sitios HTML de ejemplo 39–42
- creación de bots 260
- creación de bots con LUIS 264–267  
  descripción general de 34–35  
  ejecución de bots con LUIS 264–267  
  implementación de la aplicación en la aplicación web que ejecuta varias instancias 140  
  lenguajes y entornos compatibles 34–35  
  ranuras de implementación y 35, 44–46  
  registros de diagnóstico, visualización 42–44  
Replicación de Azure a Azure 203  
transmisión de datos de Azure IoT Hub a 309–315
- 
- B**
- bases de datos  
  escalamiento 143–144  
  en Cosmos DB  
    agregar redundancia global a 149–152  
    creación 145, 149–152  
    rellenar 145–149
- Bases de datos estructurados SQL 142
- Bloque de mensajes del servidor (SMB) 53
- bloqueos 79–80
- Botón de Control de acceso (IAM) 79
- Botón Implementar en Azure 98
- bots para aplicaciones web  
  creación 260  
  desarrollo con LUIS 264–267  
  ejecución con LUIS 264–267
- 
- C**
- captura de paquetes de red 186–188
- carácter de barra invertida 40, 68
- CD (entrega continua) 75
- certificado SSL 207
- certificados 270  
  creación 229–232  
  inyección 229–232
- certificados SSL personalizados 207
- CI (integración continua) 75
- ciclos de vida de aplicaciones 76–77
- científicos de datos, herramientas para 257–259
- cifrado 206–215
- almacenamiento de claves en Azure Key Vault 211–213 de VM 211–214
- descripción general de 206–208  
en reposo 208–209  
SSE (Storage Service Encryption) 209–210
- clave de función 332
- clave privada, de par de claves SSH 71
- clave pública, de par de claves SSH 20–22, 71
- claves  
  almacenamiento de claves de cifrado en Azure Key Vault 211–213  
  creación de almacenes de claves 219–221
- CLI (interfaz de la línea de comandos) 12
- clústeres con AKS 294–295
- colecciones 146
- código de origen 5
- comando az cosmosdb show 152
- comando az group create 131
- comando az keyvault create 212
- comando az keyvault secret show 221
- comando az storage account create 210
- comando az vm create 51, 95, 197
- comando az vm disk attach 51
- comando az vm list-sizes 127
- comando az vm resize 127, 129
- comando az vm show 105
- comando az vm show 95
- comando git push azure master 156
- comando git push dev master 45
- comando install 27
- comando ssh-keygen 21
- comandos, ajuste de líneas largas 40
- condiciones de rendimiento, alertas para 181–182
- conectividad de red (vNIC) 11
- conexión de Protocolo de escritorio remoto (RDP) 20, 71
- conexión RDP (Protocolo de escritorio remoto) 20, 71
- conexiones 270
- conexiones de red de acceso remoto 68–69
- configuración
- configurar  
    sondeos de estado 110–112  
    VM con equilibradores de carga 119–122
- conjuntos de escalamiento, para VM 129–136  
  creación 131–133  
  creación de reglas de escalamiento automático 133–136
- Conjuntos de disponibilidad 91
- distribución de VM en 98–101
- Redundancia de VM con 96–102  
  dominios de actualización 97–98  
  dominios de error 96–97
- visualización de la distribución de VM en 101–102
- contenedores 146, 284–299
- ACI (Azure Container Instance) 289–292

**A**

- AKS (Azure Kubernetes Service) 293–297
  - creación de clústeres con 294–295
  - ejecución de sitios web en Kubernetes 295–297
  - descripción general de 284–288
- control
  - recursos 78–79
  - tráfico con NSG 64–68
    - asociar NSG con subredes 66–67
    - creación de NSG 64–65
    - creación de reglas de filtrado para 67–68
  - control de acceso basado en roles (RBAC) 78, 161, 211
  - copias de seguridad 191–204
    - Azure Backup 191–201
      - directivas y retención 193–196
      - Programaciones de copia de seguridad 196–198
      - restauración de VM 198–201
    - Azure Site Recovery 201–204
  - copias de seguridad incrementales 193
  - Cosmos DB 141–157
    - acceso a los datos distribuidos globalmente 152–156
    - agregar redundancia global a 149–152
    - creación de cuentas y bases de datos 145–152
    - creación y relleno de bases de datos 145–149
    - descripción general de 141–144
      - escalamiento de bases de datos 143–144
      - bases de datos (NoSQL) no estructuradas 142–143
      - bases de datos estructuradas (SQL) 142
    - Implementación de la aplicación web con 156–157
  - CPU virtual (vCPU) 11
  - Creación de la Web de las Cosas (Guinard and Trifa) 315
  - credenciales 270
  - Cuenta de Azure, creación de 5–7
  - cuentas
    - en Azure Automation, creación de 271–272
    - en Cosmos DB
      - agregar redundancia global a 149–152
      - creación 145–149, 152
      - rellenar 145–149
    - Cuentas de ejecución 272
    - cuotas predeterminadas 102
    - cuotas 102, 132

**D**

  - datos distribuidos globalmente 152–156
  - datos estructurados 144
  - datos no estructurados 144
  - DC/OS (sistema operativo del centro de datos) 293
  - DDoS (denegación de servicio distribuido) 182
  - delegación de dominios reales 160–162
  - denegación de estado 238
  - denegación de servicio distribuido (DDoS) 182
  - dependencias 82
  - dependsOn 104
  - detección del punto de conexión 153
  - diagnóstico de arranque 175–177
  - Direcciones IP privadas 108

- direcciones IP públicas 20, 62–64, 94, 108
- Direcciones IPv4 109
- Direcciones IPv6 109
- directiva de caché de solo lectura 50
- directivas 193–196
  - RPO (objetivo de punto de recuperación) 194–195
  - RTO (objetivo de tiempo de recuperación) 195–196
- disco duro virtual (VHD) 53
- discos
  - agregado a VM 50–52
  - discos de datos 49–50
    - opciones de almacenamiento en caché 50
    - temporales 49–50
  - discos administrados 18
  - discos de datos 49–50
  - discos duros estándar (HDD) 18
  - Discos SSD (unidad de estado sólido) premium 18–19
  - discos temporales 49–50
- DKIM (DomainKeys Identified Mail) 160
- Docker 284, 287
- Docker Swarm 293
- Dockerfiles 291–292
- DomainKeys Identified Mail (DKIM) 160
- dominios
  - actualización de 97–98
  - falla de 96–97
    - real, delegación a Azure DNS 160–162
  - dominios de actualización 96
  - dominios de error 96–97
- DR (recuperación ante desastres) 201
- DSC (Desired State Configuration) 179, 278, 282–283
- DSC (Desired State Configuration) de PowerShell 278–282
  - definición 280–282
  - Servidores de extracción de Azure Automation y 179, 280–282
- DSVM (máquinas virtuales de ciencia de datos) 258

## E

---

- Editor de Visual Studio 85–86
- Ejemplo de Google Maps 256
- eliminación de VM protegidas 205
- enrutamiento de rendimiento 163–164
- enrutamiento de tráfico directo con reglas de traducción de direcciones de red 114–116
- enrutamiento geográfico 163–164
- enrutamiento global, con Traffic Manager 162–173
  - creación de perfiles de Traffic Manager 164–166
    - distribución global del tráfico a la instancia más cercana 167–173
  - entidad de servicio 222
  - entornos aislados 36
  - Entornos de App Service 36
  - entrega continua (CD) 75
  - equilibrador de carga interna 108
  - equilibrador de carga interno 108

equilibradores de carga 94  
 componentes de 106–119  
 asignación de grupos de VM a grupos de back-end 116–119  
 creación de grupos IP de frontend 108–110  
 definición de la distribución de tráfico con reglas del equilibrador de carga 112–114  
 enrutamiento de tráfico directo con reglas de traducción de direcciones de red 114–116  
 sondeos de estado 110–112  
 creación y configuración de VM con 119–122  
 definición de la distribución de tráfico con reglas 112–114  
 en acción 120–122  
 errores de autenticación 332  
 escalamiento  
   bases de datos 143–144  
   horizontal de recursos 128–129  
   reducir VM 127  
   VM verticalmente 125–127  
   redimensión de VM 126–127  
   reducción vertical 127  
 Web Apps  
   descripción general de 136–139  
   verticalmente 127–128  
 Estándar federal de procesamiento de información (FIPS) 218  
 ETW (seguimiento de eventos para Windows) 180  
 Extensión de script personalizada 179  
 extensiones 305

## F

filtrado 67–68  
 FIPS (Estándar federal de procesamiento de información) 218  
 flujos IP, verificación 183–184  
 foro, para este libro 5  
 FQDN (nombre de dominio completo) 63  
 función concat, administrador de recursos 85  
 función copy, Administrador de recursos 84  
 función copyIndex() 84, 98, 102, 104  
 Function Apps 328–331

## G

Gardner, Lyza Danger 315  
 Git 12  
   aprendizaje 37  
   contraseña para, restablecimiento 314  
   implementación de sitios HTML de ejemplo con 39–42  
 GitHub  
   Azure Automation y control de origen con 274  
   cuenta para, creación 7

descripción general de 39  
 Ejemplos de arranque rápido de Azure en 87  
 recursos 333  
 repositorio de este libro 5  
 GPU (unidad de procesamiento gráfico) 267  
 GRS (almacenamiento con redundancia geográfica) 56  
 grupo IP de back-end, en equilibradores de carga 107  
 grupos  
   back-end 116–119  
   grupos IP de front-end 108–110  
   grupos de back-end 107, 116–119  
   grupos de recursos 315  
   grupos de seguridad de red. Consulte NSG  
   grupos IP de front-end 107–110  
   Grupos IP 107  
 Guinard, Dominique D. 315

## H

HashiCorp 86  
 herramientas de terceros 86  
 Horario universal coordinado (UTC) 196  
 host bastión 23–24  
 HPC (informática de alto rendimiento) 267  
 HSM (módulos de seguridad de hardware) 212, 217–218  
 HTTP 20, 168, 206  
 HTTPS 20, 168, 206  
 Hyper-V 15

## I

IA (inteligencia artificial) 253–268  
 Azure Cognitive Services 259–260  
 Bots de Aplicaciones web  
   creación 260  
   desarrollo con LUIS 264–267  
   ejecución con LUIS 264–267  
   descripción general de 254–255  
 LUIS 261–264  
   machine learning y 254–259  
 IaaS (Infraestructura como servicio) 9, 14, 33–34  
 IaC (infraestructura como código) 82  
 identidades administradas asignadas por el sistema 222  
 identidades administradas asignadas por el usuario 222  
 IIS (Internet Information Services) 29, 233  
 Imágenes de VM 16–17  
 IMDS (Instance Metadata Service) 222  
 implementación de sitios HTML 39–42  
 informática sin servidor 317–333  
   creación de aplicaciones de función para analizar datos de dispositivos de IoT 328–331  
   creación de aplicaciones lógicas 325–328  
   plataformas de mensajes 319–325  
     Azure Event Grid 320–321

Azure Event Hubs 321–322  
 Azure Service Bus 321–322  
 creación de service bus 322–325  
 descripción general de 317–319  
 integración de Service Bus con IoT hubs 322–325  
 plataformas de mensajes 319–325  
 Recursos de GitHub 333  
 infraestructura como código (IaC) 82  
 Infraestructura como servicio (IaaS) 9, 14, 33  
 instalación de servidores web 24–27  
 Instance Metadata Service (IMDS) 222  
 instancias, creación 290–292  
 integración continua (CI) 75  
 Intercambio automático 46  
 intercambio con vista previa 46  
 interfaz de la línea de comandos (CLI) 12  
 Internet Information Services (IIS) 29, 233  
 intervalo de sondeo del punto de conexión 168  
 intervalos de direcciones IP 60  
 inyección de certificados 229–232

**J**

JavaScript en las Cosas (Gardner) 315  
 JSON (Notación de objetos JavaScript) 82–83, 86  
 JWT (JSON Web Token) 226

**K**

Kubernetes 293, 295–299  
 Consulte también AKS  
 Kubernetes en acción (Luksa) 298  
 inteligencia artificial. Consulte IA

**L**

Language service 259  
 LCM (Administrador de configuración local) 278  
 Lenguaje de consulta estructurada (SQL) 53, 142  
 Lenguaje de programación de Python 28, 34  
 lenguaje de programación Perl 34  
 lenguajes compatibles 34–35  
 Linux  
   ejecución de Web Apps en 34  
   uso de DSC con 282–283  
 LRS (almacenamiento redundante a nivel local) 56  
 LTS (soporte a largo plazo) 22  
 LUIS (Language Understanding Intelligent Service)  
   creación de bots de aplicaciones web con 264–267  
   descripción general de 257–264  
   ejecución de bots de aplicaciones web con 264–267  
 Luksa, Marko 298

**M**


---

machine learning. Consulte ML  
 máquinas virtuales de ciencia de datos (DSVM) 258  
 máquinas virtuales. Consulte VM  
 Marco de protección del remitente (SPF) 160  
 Marketplace, Azure 7  
 materiales de aprendizaje 333  
 Maven 12  
 memoria (vRAM) 11  
 mensaje de error 31  
 Message Analyzer, Microsoft 186  
 Método de enrutamiento de prioridad, Traffic Manager 163  
 Método de enrutamiento ponderado, Traffic Manager 163  
 métricas de rendimiento 178–182, 188  
 Microsoft's Message Analyzer 186  
 ML (machine learning) 253–268  
   Azure Cognitive Services 259–260  
   Bots de Aplicaciones web  
     creación 260  
     desarrollo con LUIS 264–267  
     ejecución con LUIS 264–267  
   descripción general de 255–256  
   herramientas para científicos de datos 257–259  
   inteligencia artificial y 254, 256  
   LUIS (Language Understanding Intelligent Service) 261–264  
     relación con la inteligencia artificial 257–259  
   Modo Aplicar y autocorregir, DSC 279  
   Modo Aplicar y supervisar, DSC 279  
   modo basado en puerto, sondeos de estado 110  
   modo basado en ruta HTTP, sondeos de estado 110  
   modo de afinidad de sesión 112–113  
   Modo Solo aplicar, DSC 279  
   módulo nx 283  
   módulos 270  
   MSI (identidades de servicio administrado) 221–229

**N**


---

NAT (Traducción de direcciones de red) 107, 114–116  
 navegadores web, creación de VM desde 22  
   Azure Storage 18  
   Tamaños de las VM 17  
 Network Watcher 184  
 NIC (tarjetas de interfaz de red) 61, 117  
 Nivel de grupo de seguridad de aplicaciones 185  
 Nivel de NIC virtual 185  
 Nivel de subred 185  
 NoSQL (bases de datos no estructurados) 142–143  
 NSG (grupos de seguridad de red) 20  
   asociar con subredes 66–67  
   creación 64–65, 118  
   creación de reglas de filtrado 67–68  
   descripción general de 112

en Azure Security Center 234  
protección y control de tráfico con 64–68  
visualización de reglas efectivas 184–186  
nubes, protección de la información en 216–221  
almacenes de software y HSM 217–218  
creación de almacenes de claves y secretos 219–221

## O

objetivo de tiempo de recuperación (RTO) 193  
OMS (Operations Management Suite) 243, 272  
Opción de Prueba en chat web 266  
Opción Desmontar discos 199  
orquestador de contenedor 293

## P

PaaS (Plataforma como servicio) 10, 33, 37, 137  
paquetes de red 186–188  
parámetro -A 121  
parámetro de intervalo, sondeos de estado 111  
parámetro de umbral, sondeos de estado 111  
parámetro enableHttpsTrafficOnly 210  
parámetro -no-self-perms 220  
parámetro -zone 95  
parámetros 82, 84, 89  
pares de claves 20  
pares de claves SSH 20–22  
profiles, en Traffic Manager 164–166  
PHP 34  
Plan de servicio básico 36  
plan de servicio estándar 36  
Plan de servicio gratuito/compartido 36  
Plan de servicio premium 36  
Planes de servicio de aplicaciones 35–38  
planes de servicio para aplicaciones 35–38  
plantillas de inicio rápido de Azure 7  
plantillas, para Azure Resource Manager 81–87  
    almacenamiento 87  
    creación 82–84  
    creación de múltiples tipos de recursos 84–85  
    herramientas para crear 85–86  
plataforma Azure  
    almacenamiento en 18  
    descripción general de 8–13  
    herramientas de administración 11–13  
        Azure CLI local 13  
        Azure Cloud Shell 12–13  
        Azure Portal 12  
        Azure PowerShell 13  
    solución de problemas 31–32  
    virtualización en 10–11  
Plataforma como servicio (PaaS) 10, 33  
plataformas de mensajes 319–325  
    Azure Event Grid 320–321

Azure Event Hubs 321–322  
Azure Service Bus 321–322  
creación de service bus 322–325  
integración de Service Bus con IoT hubs 322–325  
PowerShell. Consulte Azure PowerShell  
preparación 41  
programaciones 134, 270  
Programaciones de copia de seguridad 196–198  
Propiedad de Mensaje de texto 56  
protección  
    recursos 78–79  
    tráfico con NSG 64–68  
        asociar NSG con subredes 66–67  
        creación de NSG (grupos de seguridad de red) 64–65  
        creación de reglas de filtrado para 67–68  
protección de recursos 79–80  
protocolo de supervisión de punto de conexión 168  
punto de conexión de eventos 310, 312  
punto de recuperación 193  
Puntos de conexión 323  
puntos de conexión de servicio 146

## R

RA-GRS (almacenamiento con redundancia geográfica con acceso de lectura) 57  
ranura de producción 46  
ranuras de implementación 44–46  
Raspberry Pi 306–309  
RBAC (controles de acceso basado en roles) 78, 161, 184, 211  
readLocations 153  
receptores 180  
recuperación ante desastres (DR) 201  
recursos 5, 7  
    escalamiento horizontal 128–129  
    con etiquetas  
        administración 80–81  
        agrupación 80–81  
    control 78–79  
    limpieza 30  
    protección 78–79  
        protección con bloqueos 79–80  
    recursos de red 94–95  
    red anycast 160  
    Redes de Azure 58–72  
        compilación de aplicaciones web de ejemplo con tráfico seguro 68–72  
        creación de conexiones de red de acceso remoto 68–69  
        creación de VM 69–70  
        uso de agentes de SSH para conectarse a VM 70–72  
    componentes de las redes virtuales 58–64  
        creación de redes virtuales 59  
        creación de subredes 59  
    direcciones IP públicas 62–64

resolución DNS 62–64  
 tarjetas de interfaz de red virtuales 61  
 protección y control de tráfico con NSG 64–68  
 asociar NSG con subredes 66–67  
 creación de NSG 64–65  
 creación de reglas de filtrado para 67–68  
 redes privadas virtuales (VPN) 19, 36, 38  
 redes virtuales 58–64  
 creación 59  
 creación de subredes 59  
 direcciones IP públicas 62–64  
 resolución DNS 62–64  
 tarjetas de interfaz 61  
 redes. Consulte Redes de Azure  
 redimensión de VM 126–127  
 redundancia  
 beneficios de 90–91  
 de VM con conjuntos de disponibilidad 96–102  
 descripción general de 56–57  
 redundancia de infraestructura con zonas de disponibilidad 95  
 creación de recursos de red en zonas de disponibilidad 94–95  
 creación de VM en zonas de disponibilidad 95  
 redundancia global 149–152  
 redundancia. Consulte también redundancia de infraestructura, con Zonas de disponibilidad  
 registros de alias 160  
 registros de diagnóstico 42–44  
 Registros de host IPv4 160  
 Registros de host IPv6 160  
 registros de inicio de autoridad (SOA) 160  
 registros de puntero 160  
 registros de servicio 160  
 registros de servidor de nombres 160  
 registros. Consulte registros de diagnóstico  
 Regla AllowAzureLoadBalancerInBound 67  
 Regla AllowVnetInBound 67  
 regla default-allow-ssh 241  
 Regla DenyAllInBound 67, 184  
 reglas de escalamiento automático 133–136  
 Reglas DenyAll 185  
 relleno de bases de datos 145–149  
 remotos 41  
 reposo de datos 208  
 Resolución DNS 62–64, 158  
 resolución, con Traffic Manager 162–173  
 creación de perfiles de Traffic Manager 164–166  
 distribución global del tráfico a la instancia más cercana 167–173  
 REST (transferencia de estado representacional) 31  
 restauración de máquinas virtuales 198–201  
 restauración completa de VM 199–201  
 restauración en el nivel de archivo 199  
 restauración en el nivel de archivo 199  
 retención 193–196  
 RPO (objetivo de punto de recuperación) 194–195

RTO (objetivo de tiempo de recuperación) 195–196  
 revisión de actualizaciones 245–249  
 Rol de administrador de acceso de usuario 78  
 Rol de colaborador 78  
 Rol de colaborador de máquina virtual 79  
 rol de colaborador de sitio web 79  
 rol de lector 78  
 Rol de propietario 78  
 RPO (objetivo de punto de recuperación) 193–195  
 RTO (objetivo de tiempo de recuperación) 193, 195–196  
 runbooks, para Azure Automation 274–277  
 descripción general de 272–274  
 ejecución 276–277  
 ejecutar 182  
 visualización de resultados de 276–277

## S

---

SaaS (software como servicio) 10  
 secretos  
 creación 219–221  
 obtención desde las VM con MSI 224–229  
 seguridad 115  
 separación de roles 62  
 Service Bus  
 creación 322–325  
 Integración con IoT Hubs 322–325  
 servicePrincipalName 224  
 Servicio Azure Machine Learning 258  
 Servicio Bing Autosuggest 259  
 Servicio Bing Custom Search 259  
 Servicio Computer Vision 259  
 Servicio Content Moderator 259  
 servicio de búsqueda 259  
 Servicio de Decisión 259  
 Servicio de Voz 259  
 Servicio Face 259  
 Servicio personalizado 259  
 Servicio Speaker Recognition 259  
 Servicio Translator Text 259  
 Servicio Vision 259  
 servicios con redundancia de zona 93  
 servicios zonales 93  
 Servidor web LAMP 27, 72  
 servidores de bases de datos, escala vertical para 126  
 servidores de extracción 280–282  
 servidores web  
 en acción 28–29  
 instalación 24–27  
 Shell Bash 12  
 símbolo de separación 27  
 sistema de numeración basado en cero 99  
 sistema operativo del centro de datos (DC/OS) 293  
 sistemas de numeración, basado en cero 99  
 sitios HTML, implementación 39–42

sitios web, ejecución en Kubernetes 295–297  
SLA (acuerdos de nivel de servicio) 247  
SMB (bloque de mensajes del servidor) 53  
Software como servicio (SaaS) 10  
solicitud de curl 226–227  
solución de problemas 175  
  alertas 178–182  
  Azure Network Watcher 182–188  
    captura de paquetes de red 186–188  
    verificación de flujos de IP 183–184  
    visualización de reglas efectivas de NSG 184–186  
Diagnóstico de VM 175–177  
  métricas de rendimiento 178–182  
plataforma Azure 31–32  
sondeos de estado  
  configuración 110–112  
  creación 110–112  
  descripción general de 107  
SONiC (software para Redes abiertas en la nube) 11  
Soporte a largo plazo (LTS) 22  
SPF (marco de protección del remitente) 160  
SQL (lenguaje de consulta estructurada) 53, 142  
SSD de alto rendimiento 18  
SSD estándar 18–19  
SSE (Storage Service Encryption) 209–210  
SSH (Shell de socket seguro)  
  agentes para conectarse a las VM 70–72  
  conexión a VM con 24–27  
subredes  
  asociación de NSG con 66–67  
  creación 59  
sucursales, en Git 41  
supervisión 175  
  alertas 178–182  
  Azure Network Watcher 182–188  
    captura de paquetes de red 186–188  
    verificación de flujos de IP 183–184  
    visualización de reglas efectivas de NSG 184–186  
Diagnóstico de VM 175–177  
  métricas de rendimiento 178–182

## T

Tamaños de las VM de GPU 17  
tamaños de VM de uso general 17  
tamaños de VM optimizadas para almacenamiento 17  
tamaños de VM optimizadas para memoria 17  
tamaños de VM optimizadas para procesamiento 17  
tarjetas de interfaz de red (NIC) 61  
tarjetas de interfaz 61  
Terraform 86  
Tiempo de vida (TTL) 167  
tipos de recursos 84–85  
Tipos de registros de Azure DNS 160  
token de firma de acceso compartido (SAS) 87  
token SAS (firma de acceso compartido) 87

Traffic Manager  
  Área de solución de problemas 183  
  creación de perfiles en 164–166  
  distribución global del tráfico a las instancias más cercanas 167–173  
  enrutamiento global y resolución con 162–173  
  Implementación de aplicaciones web en 174  
  tráfico  
    definición de la distribución de tráfico con reglas del equilibrador de carga 112–114  
    distribución global a instancias más cercanas 167–173  
    enrutamiento de tráfico directo con reglas de traducción de direcciones de red 114–116  
    protección y control de tráfico con NSG 64–68  
      asociar NSG con subredes 66–67  
      creación de NSG 64–65  
      creación de reglas de filtrado para 67–68  
  tráfico de red  
    administración 158–174  
    enrutamiento 158–174  
  tráfico directo, enrutamiento 114–116  
  tráfico seguro, compilación de aplicaciones web con 68–72  
    creación de conexiones de red de acceso remoto 68–69  
    creación de VM 69–70  
    uso de agentes de SSH para conectarse a VM 70–72  
  tráfico web  
    creación de reglas que permitan 28  
    permiso para llegar a las VM 27–29  
Transferencia de estado representacional (REST) 31  
transmisión de archivos de registro 43  
transmisión de datos de IoT Hub 309–315  
Trifa, Vlad M. 315  
TTL (tiempo de vida) 167

## U

---

ubicaciones del punto de conexión 153  
Ubuntu Linux 14, 26  
Umali, Rick 37  
UTC (horario universal coordinado) 196  
Utilidades de DevOps de Azure 7

## V

---

Vamos a cifrar el proyecto 207  
variable access\_token 226  
variable connectionString 307  
variable database\_password 228  
variable iotconnectionstring 312  
variables 82, 84, 89, 271  
Ventana Descripción general de Security Center 236  
Ventana Información general de Update Management 243

- verificación de flujos de IP 183–184  
 VHD (disco duro virtual) 53  
 virtualización 10–11  
 VM (máquinas virtuales)  
   escalamiento vertical 125–127  
   agregar discos a 50–52  
   almacenamiento 47–50  
     almacenamiento estándar versus almacenamiento premium 48–49  
       discos de datos 49–50  
       discos temporales 49–50  
       opciones de almacenamiento en caché de disco 50  
   asignación de grupos de a grupos de back-end 116–119  
   cifrado de 211–214  
     almacenamiento de claves de cifrado en Azure Key Vault 211–213  
       laboratorio 214–215  
   conexión a 120–122  
     con agentes de SSH 70–72  
     con SSH 24–27  
   configuración 15–20  
     Azure Storage 18–19  
     imágenes de VM y 16–17  
     redes virtuales 19–20  
     tamaños de las VM 17–18  
   configuración con equilibradores de carga 119–122  
   conjuntos de escala 129–136  
     creación 131–133  
     creación de reglas de escalamiento automático 133–136  
       Instalación de aplicaciones en 139  
       creación 14–32, 69–70  
       ahorrar costos y 18  
       con equilibradores de carga 119–122  
       desde navegadores web 22  
       en zonas de disponibilidad 95  
       limpieza de recursos 30  
       solución de problemas de Azure 31–32  
       VM de Windows 29–30  
   desasignación 30  
   diagnóstico 175–177  
   distribución en los conjuntos de disponibilidad 98–101  
   eliminación 30  
   extensiones de diagnóstico 178  
   implementación desde plantillas 102–105  
   instalación de servidores web 24–27  
   obtención de secretos con MSI 224–229  
   par de claves SSH, creación para la autenticación 20–22  
     permitir que el tráfico web llegue a 27–29  
       creación de reglas para permitir el tráfico web 28  
       visualización del servidor web en acción 28–29  
   redimensionamiento 126–127  
   reducción vertical 127  
   redundancia con conjuntos de disponibilidad 96–102  
     dominios de actualización 97–98  
     dominios de error 96–97  
   restauración 198–201  
     restauración completa de VM 199–201  
     restauración en el nivel de archivo 199  
     tamaños de 17  
     visualización de distribución en los conjuntos de disponibilidad 101–102  
   VM en serie 102  
   VM paralelas 102  
   VM protegidas, eliminación 205  
   VM serie B 18  
   VMware 15  
   volcados de memoria 180  
   VPN (redes privadas virtuales) 19, 36, 38, 183  
   VPN, ExpressRoute 19, 183
- 
- ## W
- 
- webhooks 274  
 WebSockets 312  
 Windows, ejecución de Web Apps en 34
- 
- ## Z
- 
- Zonas de disponibilidad 91  
   creación de recursos de red en 94–95  
   creación de VM en 95  
   redundancia de infraestructura con 95  
 ZRS (almacenamiento con redundancia de zona) 56



Microsoft.Source Newsletter - Inbox

Message

Microsoft

Microsoft.Source Newsletter | Issue 7

You're reading Microsoft.Source, the developer community newsletter featuring ideas and projects from your peers down the street –and around the world. If someone forwarded you this newsletter and you want to receive future editions, [sign up >](#)

[Give feedback](#) Get more of what you want in each edition.

**Featured Story**

**Vanilla JS and HTML –No frameworks, no libraries, no problem >**  
Do you know what it takes to render HTML elements without the complexity of AngularJS, React, Svelte, or Vue.js? See how to create a simple web page with pure HTML, CSS, and JS.  
Web, JavaScript, HTML

**What's New**

**Build a web experience to send GIFs to MXChip >**  
IoT, project

**The Making of Azure Mystery Mansion >**  
Game, Twine, PlayFab

**Trying to make FETCH happen >**  
Serverless, IoT, Azure Functions

**Events** [See all events](#)

**Cosmos DB Live Webcast / Online >**  
Expert-led, containers, .Net

**OpenHack Serverless / Los Angeles >**  
In-person event, serverless, hack

**Learning**

**Microsoft Ignite – Watch videos on demand >**  
Watch all keynotes, announcements, and sessions on demand

# By developers, for developers

Microsoft.Source newsletter

Get technical articles, sample code, and information on upcoming events in Microsoft.Source, the curated monthly developer community newsletter.

- Keep up on the latest technologies
- Connect with your peers at community events
- Learn with hands-on resources

