

AWS vs Azure

Jesús Rodríguez Heras
Carlos Llamas Jaén
Iván Castillo Caro
Sisic Dino

21 de mayo de 2019

La computación en la nube está a la orden del día, debido al abaratamiento de costes y la facilidad del escalado de los recursos empleados.

Vamos a comparar las dos principales plataformas, Amazon Web Services y Microsoft Azure en términos centrados a la utilización de los recursos por dispositivos del Internet de las Cosas, muy relacionado con la computación en la nube.

Primeramente comprobaremos como lanzar una máquina virtual en ambos servicios, útil para realizar los diversos cálculos que utiliza el sistema, posteriormente veremos como crear sitios o servicios web de cara a poder ser utilizados por los dispositivos para transmitir información y por último arrancaremos brokers de mensajería para la comunicación dispositivo-servidor.

Índice general

I	Introducción	7
1.	AWS	9
1.1.	<i>¿Qué es AWS?</i>	9
1.2.	<i>Historia</i>	9
1.3.	<i>Descripción</i>	10
2.	Azure	11
2.1.	<i>¿Qué es Azure?</i>	11
2.2.	<i>Historia</i>	11
2.3.	<i>Descripción</i>	11
II	Características y limitaciones	13
3.	Características y limitaciones de AWS	15
3.1.	<i>Características</i>	15
3.2.	<i>Limitaciones</i>	16
4.	Características y limitaciones de Azure	17
4.1.	<i>Características</i>	17
4.2.	<i>Limitaciones</i>	18
III	Comparación de los servicios ofrecidos por AWS y Azure	19
5.	Creación de máquinas virtuales en AWS y Azure	21
5.1.	<i>Creación de una máquina virtual en AWS</i>	21
5.2.	<i>Creación de una máquina virtual en Azure</i>	29

6. Creación de webs en AWS y Azure	35
6.1. Servicios Web	35
6.1.1. SOAP	35
6.1.2. REST	35
6.2. Creación de una web en AWS	36
6.3. Creación de una web en Azure	36
7. Creación de servicios de IoT en AWS y Azure	41
7.1. Creación de servicios IoT en AWS	42
7.2. Creación de servicios IoT en Azure	45
IV Anexo	51
8. Referencias	53

Parte I

Introducción

Capítulo 1

AWS

1.1. ¿Qué es AWS?

Amazon Web Services, en adelante AWS, es una colección de servicios de computación en la nube pública, también llamados servicios web¹.

Es usado en aplicaciones como Dropbox o Foursquare. Es una de las ofertas internacionales más importantes de la computación en la nube y compite directamente con Microsoft Azure aunque AWS es considerado como pionero en este campo.

1.2. Historia

AWS se lanza oficialmente en 2006 ofreciendo servicios en línea para otros sitios web o aplicaciones del lado del cliente.

La mayoría de estos servicios no están expuestos directamente a usuarios finales, sino que ofrecen una funcionalidad que otros desarrolladores puedan utilizar en sus aplicaciones.

El primer servicio de AWS lanzado para el uso público era Simple Queue Service (SQS). El cual es un servicio de colas de mensajes completamente administrado que permite desacoplar y ajustar la escala de microservicios, sistemas distribuidos y aplicaciones sin servidor. SQS elimina la complejidad y los gastos generales asociados con la gestión y el funcionamiento de middleware² orientado a mensajes, y permite a los desarrolladores centrarse en la diferenciación del trabajo.

Hoy en día, AWS proporciona una plataforma de infraestructura escalable, de confianza y de bajo costo en la nube que impulsa cientos de miles de negocios de 190 países de todo el mundo. Con centros de datos en Estados Unidos, Europa, Brasil, Singapur, Japón y Australia.

¹Es una tecnología que utiliza un conjunto de protocolos y estándares (como mensajería por XML) que sirven para intercambiar datos entre aplicaciones. No están vinculados a ningún sistema operativo, por lo que funcionan de forma independiente y simultánea. Los componentes de los servicios web son: SOAP, UDDI, WSDL, etc.

²Software que conecta componentes de software o aplicaciones para que puedan intercambiar datos entre ellas. Muy utilizado para soportar aplicaciones distribuidas.

1.3. Descripción

Como ya se ha comentado anteriormente, AWS es una plataforma ideal para lanzar aplicaciones y proyectos de forma distribuida y escalable.

Entre sus beneficios principales, podemos destacar las siguientes:

- **Bajo costo:** AWS ofrece precios bajos por uso, sin gastos anticipados ni compromisos a largo plazo. Solo pagas por lo que usas.
- **Agilidad y elasticidad instantánea:** AWS proporciona una infraestructura global y masiva en la nube que permite experimentar e iterar con rapidez. Puede implementar nuevas aplicaciones y aumentar su escala en cuanto crezca su carga de trabajo, o bien, reducirla en función de la demanda. Cuenta con redundancia y disponibilidad en todo momento.
- **Accesibilidad y flexibilidad:** AWS es una plataforma independiente del lenguaje y del sistema operativo.
- **Seguridad:** AWS es una plataforma tecnológica, segura y duradera que cuenta con certificaciones y auditorías reconocidas en el sector.

Capítulo 2

Azure

2.1. ¿Qué es Azure?

Compete directamente con **Amazon Web Services**. Es un conjunto de servicios de computación en la nube, lanzado por Microsoft. Tiene un gran número de servicios, y la posibilidad de usar tanto Windows como Linux en sus máquinas, haciéndolo compatible con aquellas empresas que requieran ambos sistemas en sus infraestructuras. Al igual que AWS, usa un modelo de pago por uso de los recursos.

2.2. Historia

Azure se lanza oficialmente en 2008 en el Professional Developers Conference de Los Ángeles, y se lanzó como beta. No fue hasta 2010 cuando pasó a ser un producto comercial.

Durante esta versión Beta que duró casi 2 años, se ofrecieron unos primeros servicios tales como SQL Database Relational, PHP, Java, etc. Posteriormente, ya en versión comercial, se han ido incluyendo características como máquinas virtuales, mejoras en SQL, escalados automáticos, Python, etc.

2.3. Descripción

Dentro de Azure podemos destacar los siguientes puntos:

- **Proceso:** El servicio de proceso de Windows Azure ejecuta aplicaciones basadas en Windows Server. Estas aplicaciones se pueden crear mediante .NET Framework en lenguajes como C# y Visual Basic, o implementar sin .NET en C++, Java y otros lenguajes.
- **Almacenamiento:** Ojetos binarios grandes (blobs) proporcionan colas para la comunicación entre los componentes de las aplicaciones de Windows Azure y ofrece un tipo de tablas con un lenguaje de consulta simple.
- **Servicios de infraestructura:** Posibilidad de desplegar de una forma sencilla máquinas virtuales con Windows Server o con distribuciones de Linux.
- **Controlador de tejido:** Windows Azure se ejecuta en un gran número de máquinas. El trabajo del controlador de tejido es combinar las máquinas en un solo centro de datos de Windows Azure formando un conjunto armónico. Los servicios de proceso y almacenamiento de Windows Azure se implementan encima de toda esta eficacia de procesamiento.

- **Red de entrega de contenido (CDN):** El almacenamiento en caché de los datos a los que se accede frecuentemente cerca de sus usuarios agiliza el acceso a esos datos.
- **Connect:** Diferentes organizaciones interactúan con aplicaciones en la nube como si estuvieran dentro del propio firewall de la organización.
- **Administración de identidad y acceso:** La solución Active Directory permite gestionar de forma centralizada y sencilla el control de acceso y la identidad. Esta solución es perfecta para la administración de cuentas y la sincronización con directorios locales.

Parte II

Características y limitaciones

Capítulo 3

Características y limitaciones de AWS

3.1. Características

Las características o servicios ofrecidos por AWS son los siguientes:



Figura 3.1: Servicios ofrecidos por AWS.

Dentro de los servicios que podemos ver en la figura 3.1, vamos a describir los más destacables:

- **Amazon Elastic Cloud (EC2):** Permite la configuración y ejecución de servidores en demanda mediante un Amazon Machine Instance (AMI).
- **AMI:** Una imagen de Amazon machine (AMI) proporciona la información necesaria para lanzar una instancia. Debe especificar una AMI al lanzar una instancia. Cuando se necesiten varias instancias con la misma configuración, se pueden lanzar desde una misma AMI.

Una AMI incluye lo siguiente:

- Una plantilla para el volumen raíz de la instancia (por ejemplo, un sistema operativo, un servidor de aplicaciones y las aplicaciones necesarias).
 - Permisos de lanzamiento que controlan qué cuentas de AWS pueden utilizar la AMI para lanzar instancias.
 - Un mapeo de dispositivos de bloques que especifica los volúmenes que se van a adjuntar a la instancia cuando se lance.
- **Amazon Simple Storage Service (S3):** Permite guardar y recuperar datos en la nube.
 - **Amazon Simple DB:** Proporciona funcionalidad de una base de datos sobre una máquina Amazon S3 basada en pares clave-valor.
 - **Amazon Simple Queue Service (SQS):** Servicio de mensajería para encolar tareas y mensajes.
 - **Amazon Relational Database Service (RDS):** Servicio web para crear, operar y escalar una base de datos en la nube.
 - **Amazon CloudFront:** Una copia de sus objetos más populares son cacheados en una red de nodos alrededor del mundo.

En cuanto al resto de servicios, AWS ofrece todo tipo de herramientas y aplicaciones para desarrolladores, empresas, Internet de las cosas (IoT), gestión de usuarios, robótica, redes y servicios multimedia.

3.2. Limitaciones

Dentro de la capa de estudiantes¹ tenemos las siguientes limitaciones (entre muchas más):

- **Amazon EC2:** Tiene límites tanto sobre el tipo de instancia que puede utilizar como la cantidad de horas que puede utilizar en un mes.
- **Amazon S3:** Tiene un límite en la cantidad de almacenamiento que puede utilizar y sobre la frecuencia con la que puede llamar a determinadas operaciones cada mes.
- **Amazon RDS:** Tiene un límite de 750 horas al mes durante los primeros 12 meses tanto para linux como para windows en cualquier combinación de instancias. Encender una máquina una vez, hará que cuente como una hora, por lo que es más recomendable iniciarla durante 3 horas (contabilizará 3 horas), que iniciarla 3 veces en una hora (también contabilizará 3 horas).

En cuanto a servicios, podemos echar en falta aquellos relacionados con la Inteligencia Artificial, o aplicaciones que hagan uso de Inteligencia Artificial en su procesamiento.

¹La que hemos podido probar gracias al acuerdo de la UCA con Amazon.

Capítulo 4

Características y limitaciones de Azure

4.1. Características

Las características o servicios ofrecidos por Azure son los siguientes:

The screenshot shows the Microsoft Azure portal interface. The left sidebar has a navigation menu with items like 'Crear un recurso', 'Inicio', 'Panel', 'Todos los servicios' (which is selected and highlighted in blue), 'FAVORITOS', and many others. The main content area is titled 'Todos los servicios' and shows a list of service categories under 'GENERAL (15)'. Each category has a count of services in parentheses. The categories listed are: COMPUTE (22), REDES (27), ALMACENAMIENTO (12), WEB (13), MÓVIL (2), CONTENEDORES (7), BASES DE DATOS (16), ANALYTICS (14), AI + APRENDIZAJE AUTOMÁTICO (8), INTERNET DE LAS COSAS (23), INTEGRACIÓN (20), IDENTIDAD (13), SEGURIDAD (8), DEVOPS (6), MIGRAR (6), ADMINISTRACIÓN Y GOBERNANZA (26), INTUNE (17), and OTRO.

Figura 4.1: Servicios ofrecidos por Azure.

Las características generales de Azure son las siguientes:

- **Autoservicio bajo demanda:** Los usuarios pueden proveerse de cómputo en la nube sin requerir interacción humana o con el mismo proveedor.
- **Acceso ubicuo a la red:** Todo lo que podamos necesitar se encuentra en la red y accesible desde la red. Disponible desde cualquier dispositivo por medio de estándares como HTML o el protocolo HTTP.

- **Agrupación de recursos independientemente de la posición geográfica:** Los recursos del proveedor se encuentran geográficamente agrupados para servir a múltiples consumidores de manera distribuida y bajo demanda.
- **Elasticidad rápida:** Las funcionalidades se proporcionan de manera muy rápida, incluso puede ser configurable para que crezca dependiendo del ambiente actual.
- **Servicio medido:** El uso de todos los recursos se puede monitorizar, lo que proporciona transparencia tanto al que expone los servicios (el proveedor) como a los que acceden a ellos (los consumidores).
- **Pago por uso:** El costo de los servicios expuestos se puede modelar con la siguiente expresión:

$$(CaracterísticasDelServicio) * (TiempoDeActividad) = CostoTotal$$

Cabe destacar, y, como se ve en la figura 4.1 que en Azure sí contamos con servicios relacionados con Inteligencia Artificial y aprendizaje automático (Machine Learning).

4.2. Limitaciones

Dentro de la capa de estudiantes¹ tenemos las siguientes **limitaciones**, de las cuales, las más destacables son (entre muchas otras):

- 750 horas de máquinas virtuales B1S² tanto para Linux como para Windows.
- 128GB de Managed Disks³ como combinación de dos discos de almacenamiento SSD de 64GB, además de 1GB en operaciones instantáneas y 2 millones de operaciones de E/S.
- 250GB de una instancia S0⁴ estándar de bases de datos SQL con 10 unidades de transacción de bases de datos.
- 1500 horas de IP dinámica para máquinas virtuales B1S.

¹La que hemos podido probar gracias al acuerdo de la UCA con Microsoft.

²Características de las máquinas B1S: 1vCPU, 1GiB de RAM y 4GiB de almacenamiento.

³Ofrece la seguridad, disponibilidad, escalabilidad y durabilidad de HDD/SSD que se necesite para todas las cargas de trabajo, desde cargas de trabajo críticas hasta escenarios de prueba.

⁴Instancia de bases de datos gestionada por Azure SQL.

Parte III

Comparación de los servicios ofrecidos por AWS y Azure

Capítulo 5

Creación de máquinas virtuales en AWS y Azure

En este capítulo se describe la creación de máquinas virtuales en los servicios de AWS y Azure para ver la escalabilidad que proporcionan dichos servicios.

Hemos seleccionado las siguientes máquinas virtuales en función del servicio:

- **AWS:** Hemos seleccionado una máquina EC2, la cual proporciona capacidad de computación escalable en la nube de AWS. El uso de Amazon EC2 elimina la necesidad de invertir inicialmente en hardware, de manera que puede desarrollar e implementar aplicaciones en menos tiempo. Puede usar Amazon EC2 para lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes y administrar el almacenamiento.
- **Azure:** Hemos seleccionado una máquina B1ls la cual es la opción ideal para servidores web pequeños, bases de datos pequeñas y entornos de desarrollo y pruebas. Ofrece una forma económica de implementar cargas de trabajo que no necesitan el uso pleno de la CPU de forma continua e irrumpen en su rendimiento.

5.1. Creación de una máquina virtual en AWS

La máquina t2.micro de Amazon EC2 seleccionada cuenta con las siguientes especificaciones técnicas:

- Un vCPU.
- 1 GB de RAM.
- 8 GB de almacenamiento (HDD o SSD).

Para crear el servicio debemos seguir los siguientes pasos:

1. Nos dirigimos a la consola de administración de AWS.

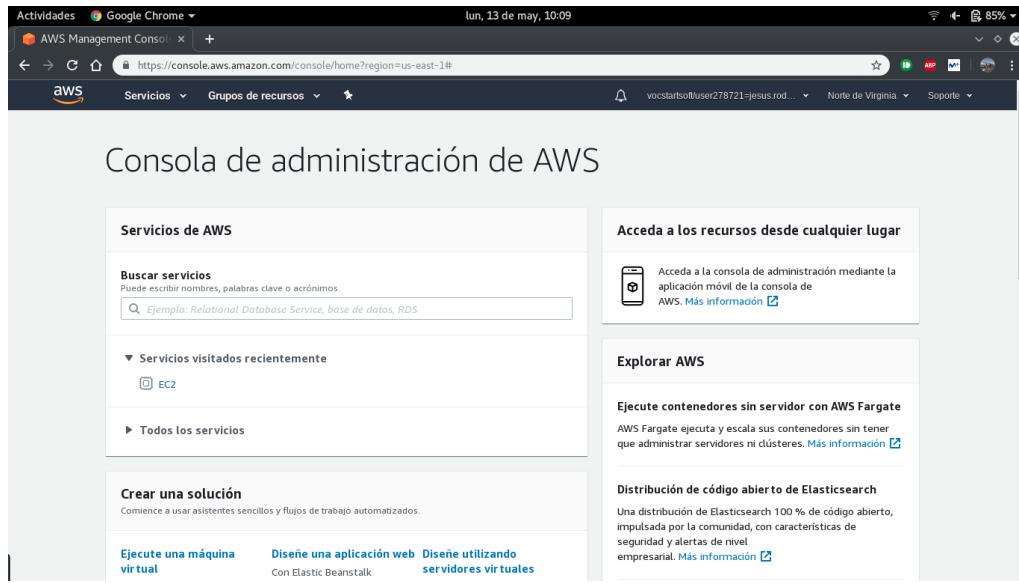


Figura 5.1: Consola de administración de AWS.

2. Seleccionamos “Ejecute una máquina virtual”.

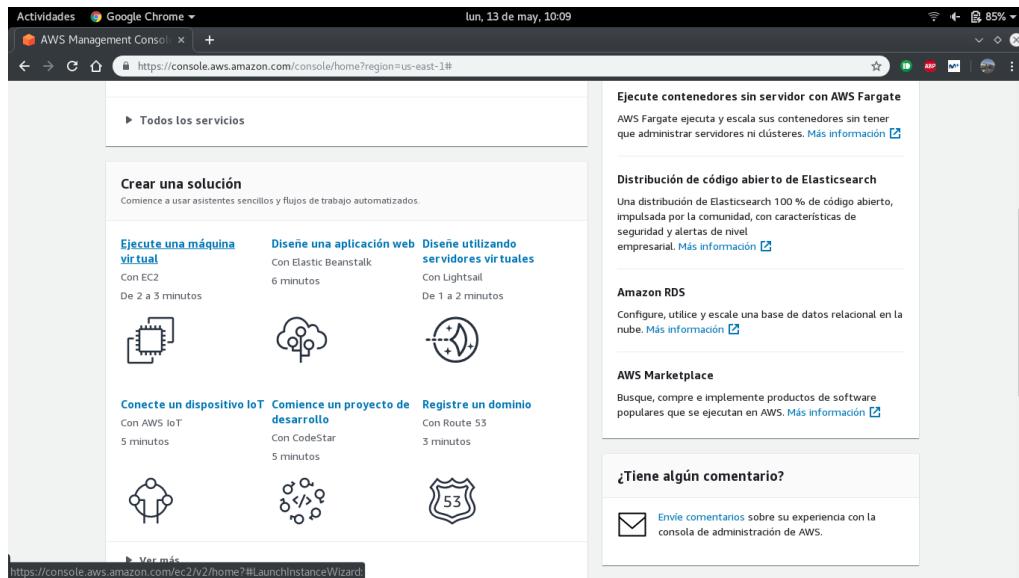


Figura 5.2: Selección de solución.

3. Seleccionamos la AMI que deseemos, en nuestro caso será “Ubuntu server 18.04 LTS (HVM), SSD Volume Type”.

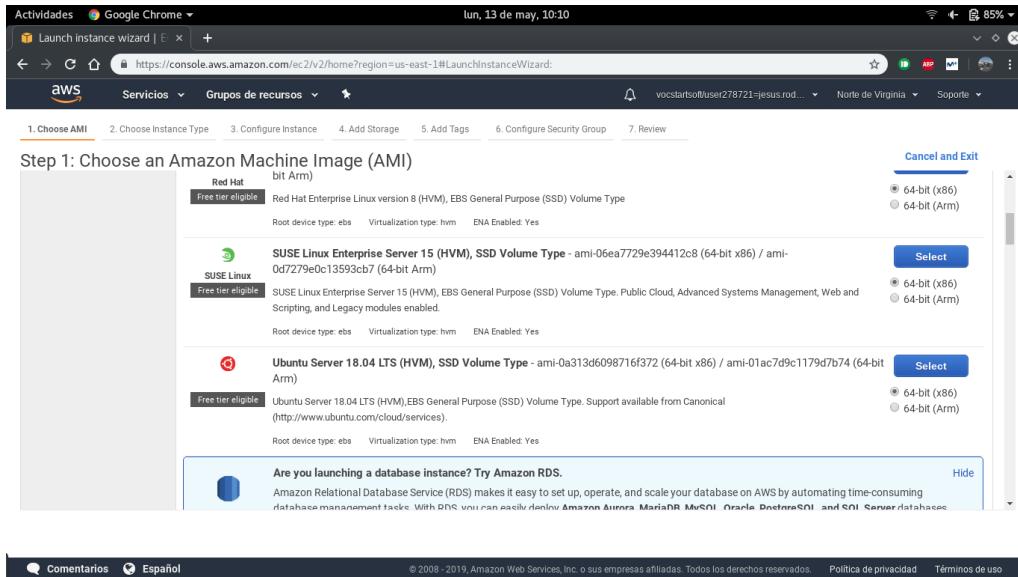


Figura 5.3: Selección de AMI.

4. Seleccionamos el tipo de instancia que queremos:

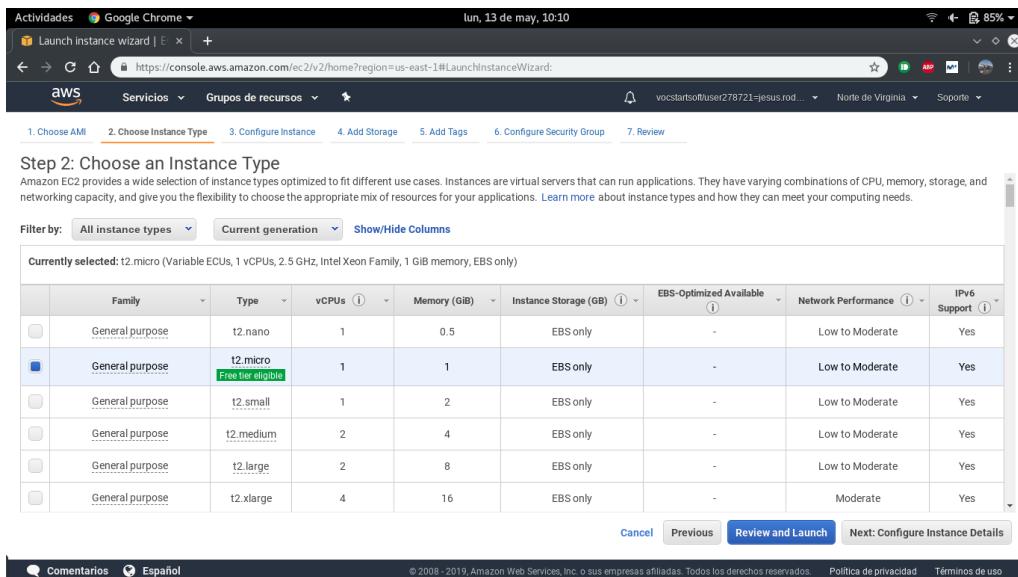


Figura 5.4: Selección de instancia.

5. Ahora debemos configurar los detalles que tendrá la instancia seleccionada:

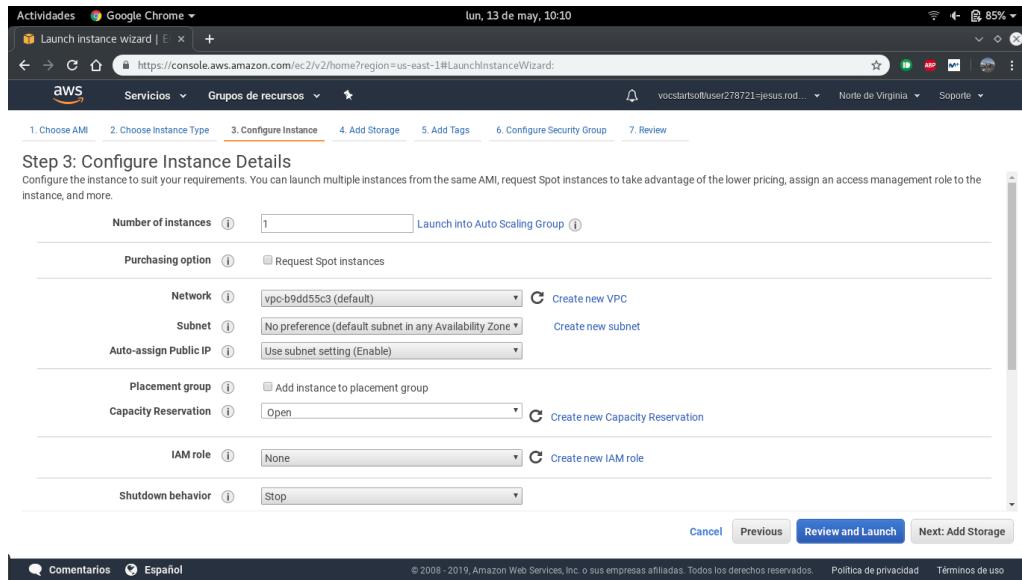


Figura 5.5: Configuración de la instancia.

6. Seleccionamos el tamaño del almacenamiento y el tipo de disco duro:

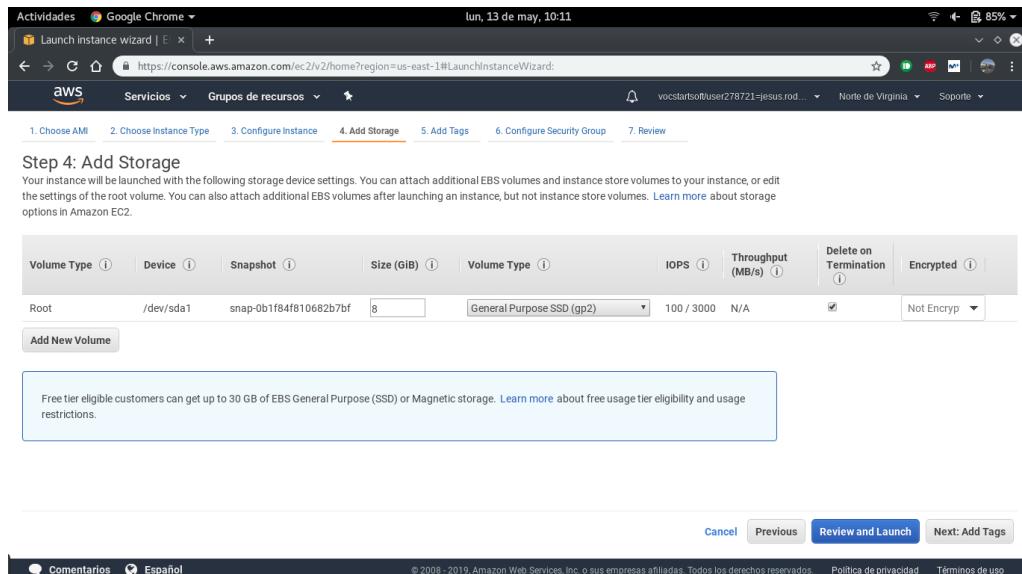


Figura 5.6: Selección de almacenamiento.

7. Añadimos las etiquetas que deseemos:

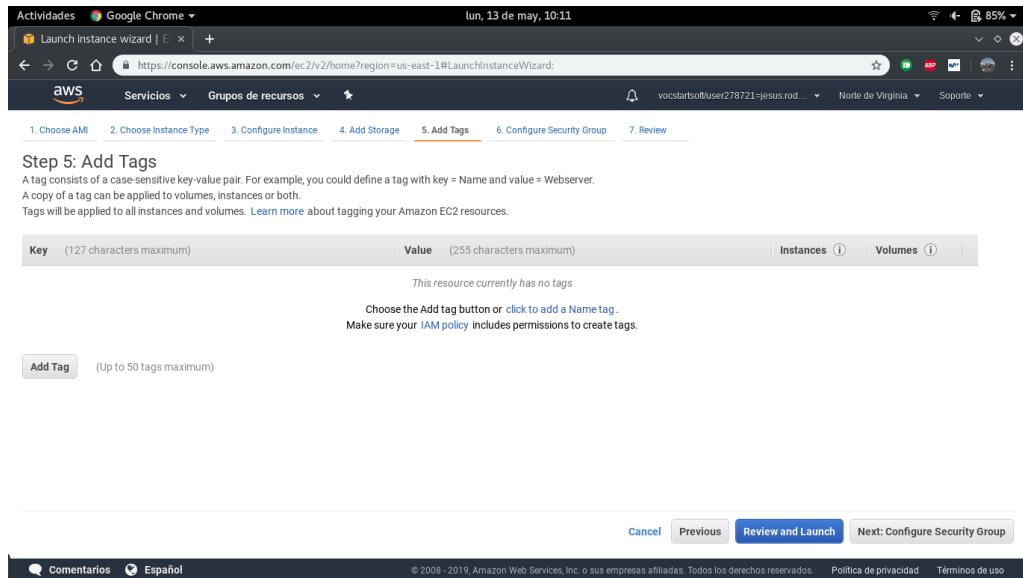


Figura 5.7: Añadimos las etiquetas.

8. Configuramos las conexiones que vayamos a tener como el SSH:

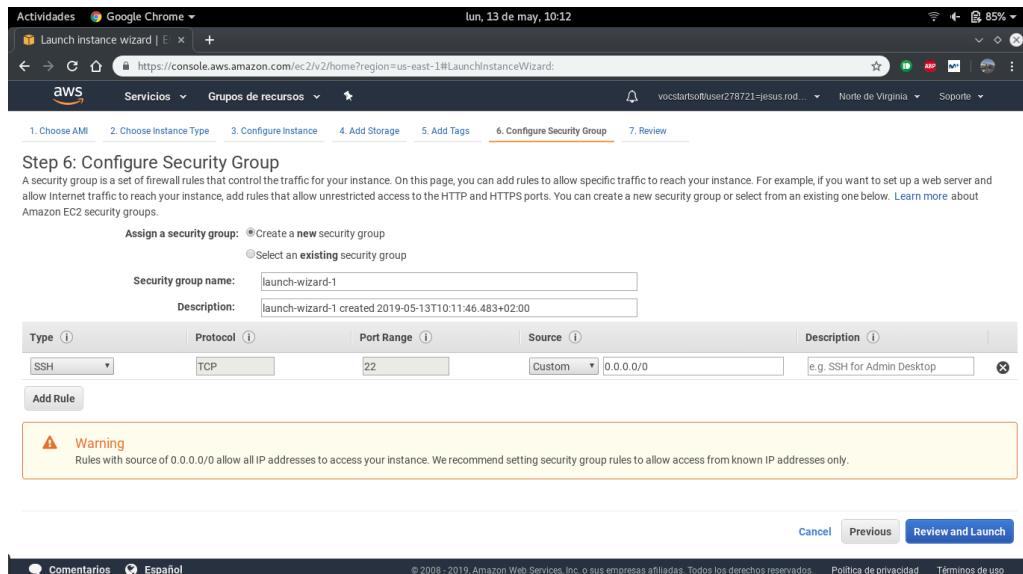


Figura 5.8: Configuración conexiones.

9. Finalmente, veremos una página con el resumen de las características que tendrá nuestra máquina:

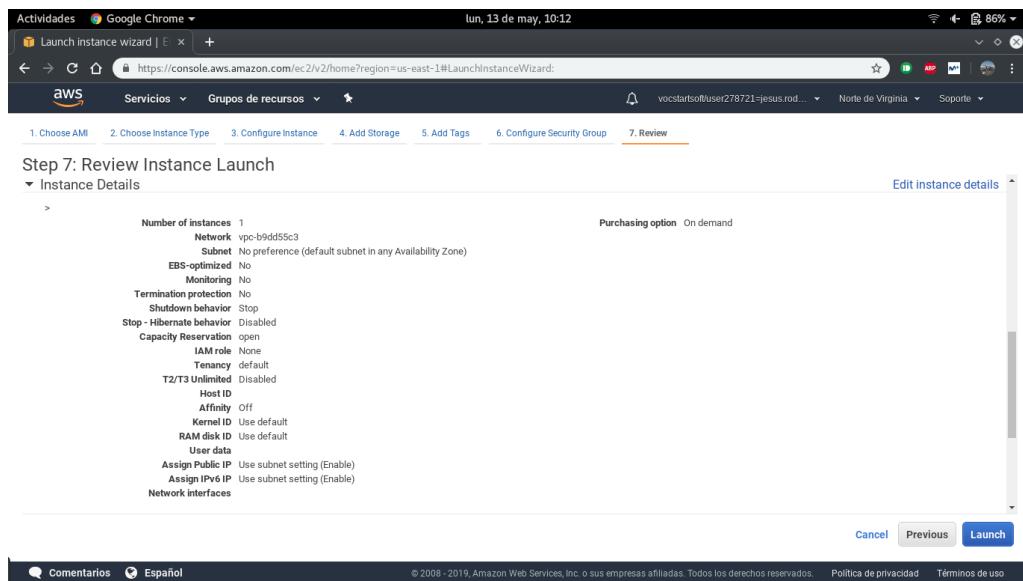


Figura 5.9: Resumen de la máquina.

10. Cuando le damos a “Launch” nos saldrá la siguiente pestaña para generar y descargar la clave SSH:

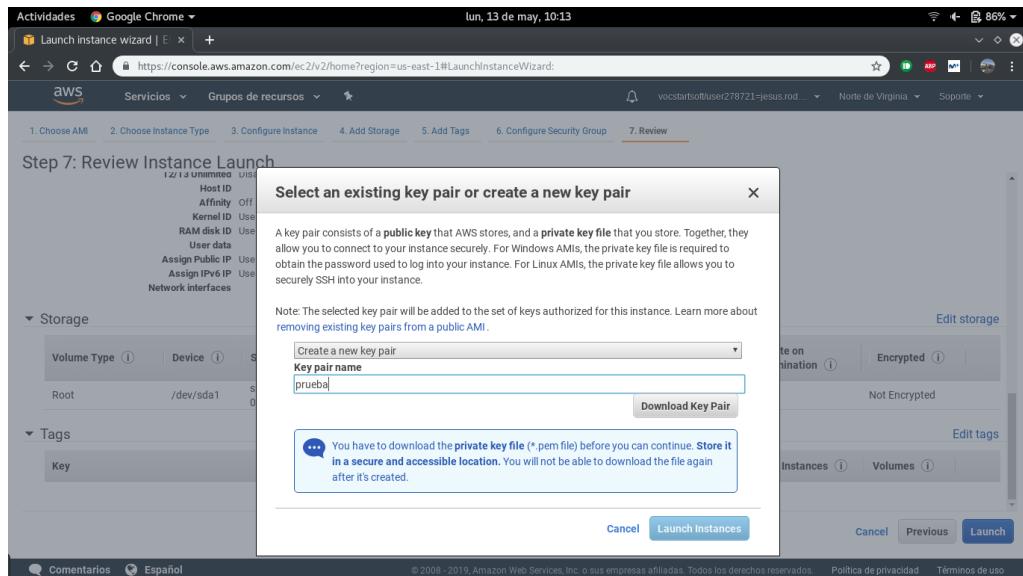


Figura 5.10: Creación de clave SSH.

11. Luego, en el apartado de “Instancias” de la consola de administración podremos ver nuestra máquina virtual ejecutándose:

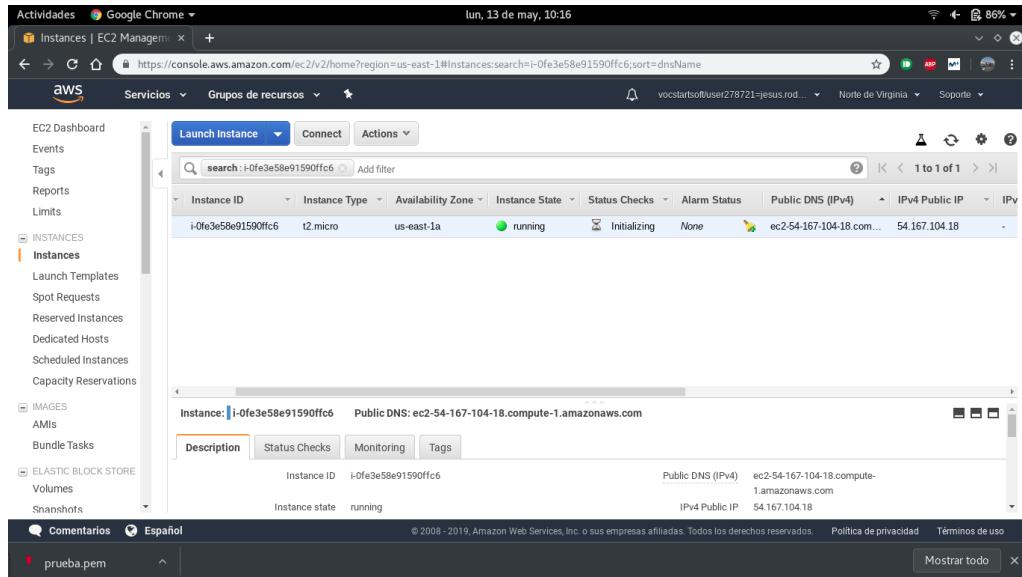


Figura 5.11: Máquina en ejecución.

12. Para conectarnos a la máquina por SSH, debemos cambiar los permisos del fichero “prueba.pem” generado e introducir el siguiente comando:

```
ssh -i prueba.pem ubuntu@54.167.104.18
```

The screenshot shows a terminal window with the following session:

```

Actividades Terminal lun, 13 de may, 10:17
ubuntu@ip-172-31-93-144: ~

Archivo Editar Ver Buscar Terminal Ayuda
This private key will be ignored.
Load key "prueba.pem": bad permissions
Permission denied (publickey).
jesus@Jesus:~/Descargas$ chmod 400 prueba.pem
jesus@Jesus:~/Descargas$ ssh -i prueba.pem ubuntu@54.167.104.18
Welcome to Ubuntu 18.04.2 LTS (GNU/Linux 4.15.0-1032-aws x86_64)

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

System information as of Mon May 13 08:16:59 UTC 2019
System load: 0.02 Processes: 86
Usage of /: 3.9% of 27.08GB Users logged in: 0
Memory usage: 14% IP address for eth0: 172.31.93.144
Swap usage: 0%

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

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.

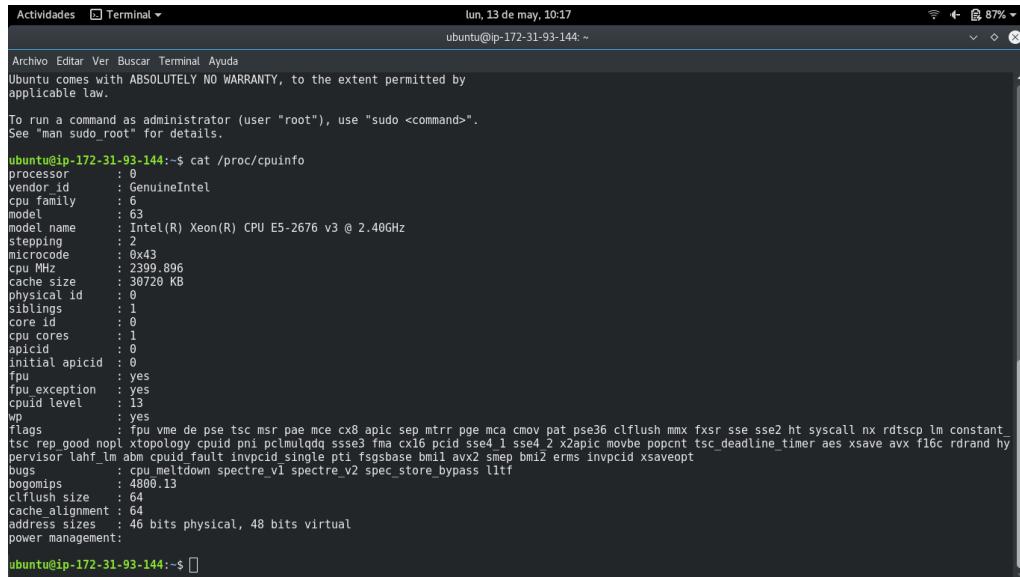
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

ubuntu@ip-172-31-93-144:~$ 
```

Figura 5.12: Conexión con la máquina virtual.

13. Si lanzamos el comando: `cat /proc/cpuinfo`, podremos ver las características del procesador de nuestra máquina virtual.



```

Actividades Terminal
lun, 13 de may, 10:17
ubuntu@ip-172-31-93-144: ~

Archivo Editar Ver Buscar Terminal Ayuda
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.

ubuntu@ip-172-31-93-144:~$ cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 63
model name    : Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz
stepping        : 2
microcode      : 0x43
cpu MHz        : 2399.896
cache size     : 30720 KB
physical id    : 0
siblings        : 1
core id        : 0
cpu cores      : 1
apicid         : 0
initial apicid : 0
fpu             : yes
fpu exception  : yes
cpuid level   : 13
wp              : yes
flags           : tsc vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ht syscall nx rdtscp lm constant_tsc rep_good nopl xtproc cpuid bmi1 pdcmulqd ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hy
pervisor lahf_lm abm cpuid_fault invpcid_single pti fsgsbase bmi1 avx2 smep bmi2 erms invpcid xsaveopt
bugs            : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass lltf
bogomips       : 4800.13
clflush size   : 64
cache alignment: 64
address sizes  : 48 bits physical, 48 bits virtual
power management:

ubuntu@ip-172-31-93-144:~$ 

```

Figura 5.13: Información de la CPU de la máquina virtual.

14. Para detener la ejecución de la máquina virtual, en la consola de administración, hacemos click derecho y seleccionamos “Stop” dentro de “Instance State”. También es posible detener la máquina virtual con el comando `sudo shutdown now`.

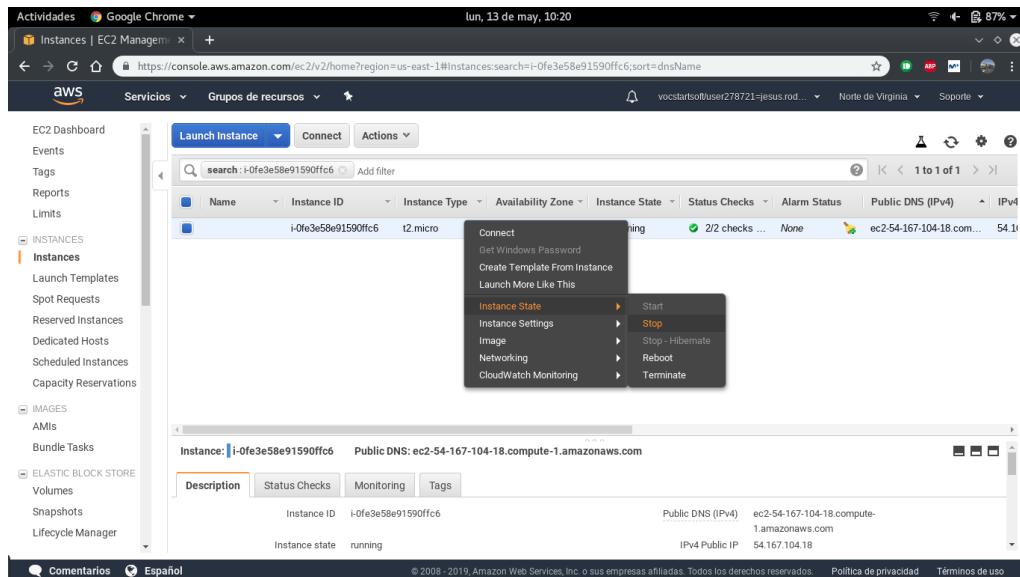


Figura 5.14: Detener la ejecución de la máquina virtual.

5.2. Creación de una máquina virtual en Azure

La máquina B1s de Azure seleccionada cuenta con las siguientes especificaciones técnicas:

- Un vCPU.
- 0.5 GB de RAM.
- 4 GB de almacenamiento (HDD o SSD).
- 200 MB de transferencia.

Para crear el servicio, debemos seguir los siguientes pasos:

1. Seleccionamos la opción “máquinas virtuales” del menú de la derecha:

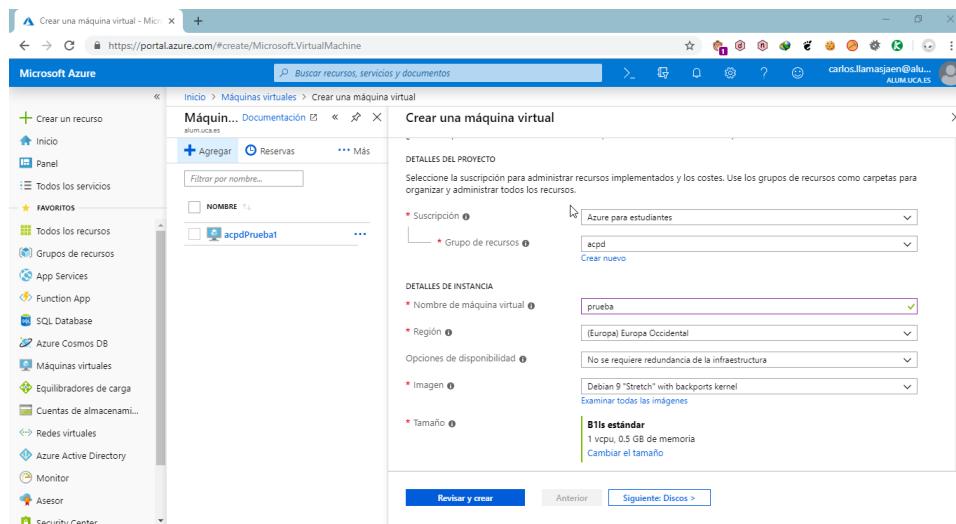


Figura 5.15: Crear máquina virtual.

2. Seleccionamos el tipo de instancia que queremos:

Seleccionar un tamaño de máquina virtual										
Explorar los tamaños de máquina virtual disponibles y sus características										
<input type="text"/> Buscar por tamaño de ...		<input type="button"/> Borrar todos los filtros								
Tamaño : Pequeño (0-4)		Generación : 2 seleccionados			Familia : Uso general		Disco Premium : Se admite		<input type="button"/> Agregar filtro	
Mostrando 11 de 230 tamaños de máquina virtual. Suscripción: Azure para estudiantes Región: Oeste de Europa Tamaño actual: Standard_D2s_v3										
TAMAÑO	OFRERA	FAMILIA	VCPU	RAM (GB)	DISCOS DE...	E/S MÁXIM...	ALMACENAMIENTO...	COMPATIBILIDAD DE...	COSTO AL MES (EST...)	
B1s	Estándar	Uso general	1	0,5	2	200	4 GB	Sí	3,76 €	
B1ms	Estándar	Uso general	1	2	2	800	4 GB	Sí	15,06 €	
B1s	Estándar	Uso general	1	1	2	400	4 GB	Sí	7,53 €	
B2ms	Estándar	Uso general	2	8	4	2400	16 GB	Sí	60,23 €	
B2s	Estándar	Uso general	2	4	4	1600	8 GB	Sí	30,12 €	
B4ms	Estándar	Uso general	4	16	8	3600	32 GB	Sí	120,46 €	
D2s_v3	Estándar	Uso general	2	8	4	3200	16 GB	Sí	75,29 €	
D4s_v3	Estándar	Uso general	4	16	8	6400	32 GB	Sí	150,58 €	
DS1_v2	Estándar	Uso general	1	3,5	4	3200	7 GB	Sí	42,66 €	
DS2_v2	Estándar	Uso general	2	7	8	6400	14 GB	Sí	85,33 €	
DS3_v2	Estándar	Uso general	4	14	16	12800	28 GB	Sí	170,66 €	

Figura 5.16: Selección de instancia.

3. Establecemos el tipo de autenticación por SSH y añadimos la clave pública:

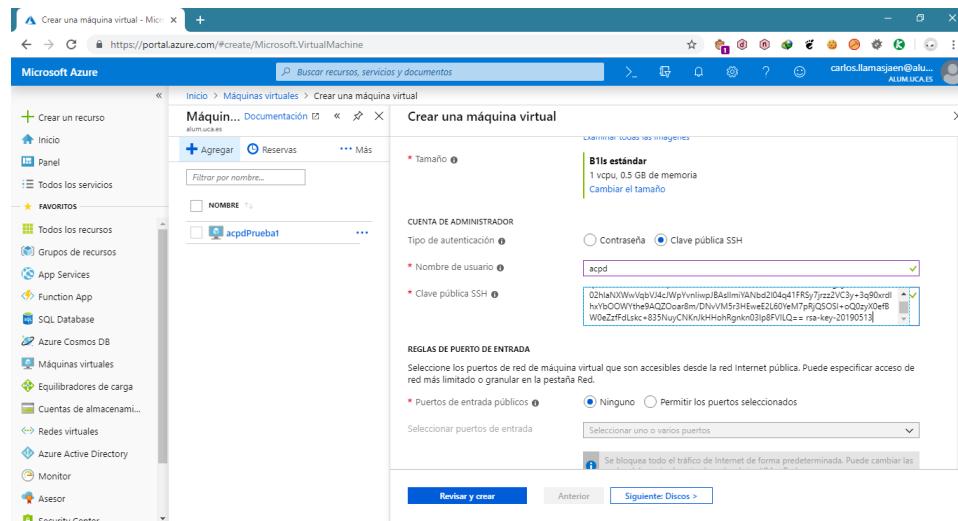


Figura 5.17: Establecer autenticación por SSH.

4. Seleccionamos el tamaño de almacenamiento:

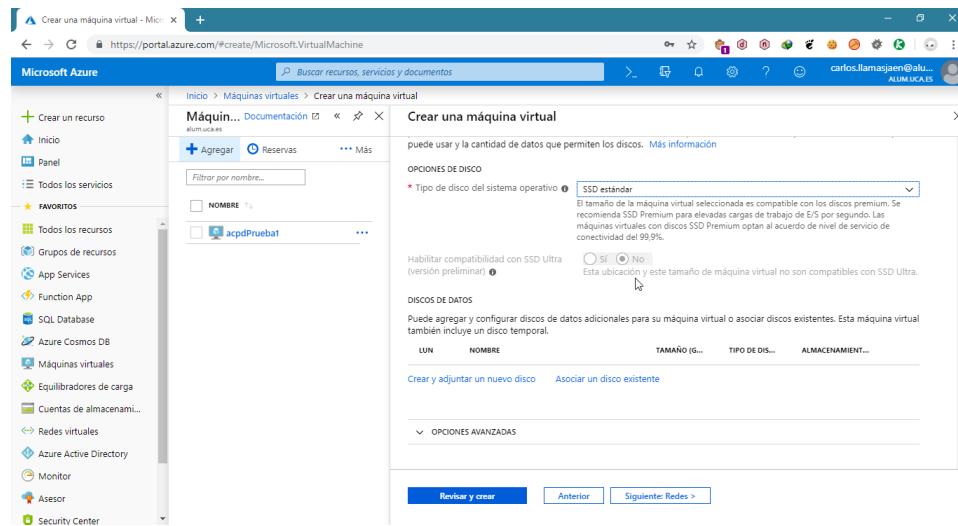


Figura 5.18: Selección de almacenamiento.

5. Configuramos las redes virtuales:

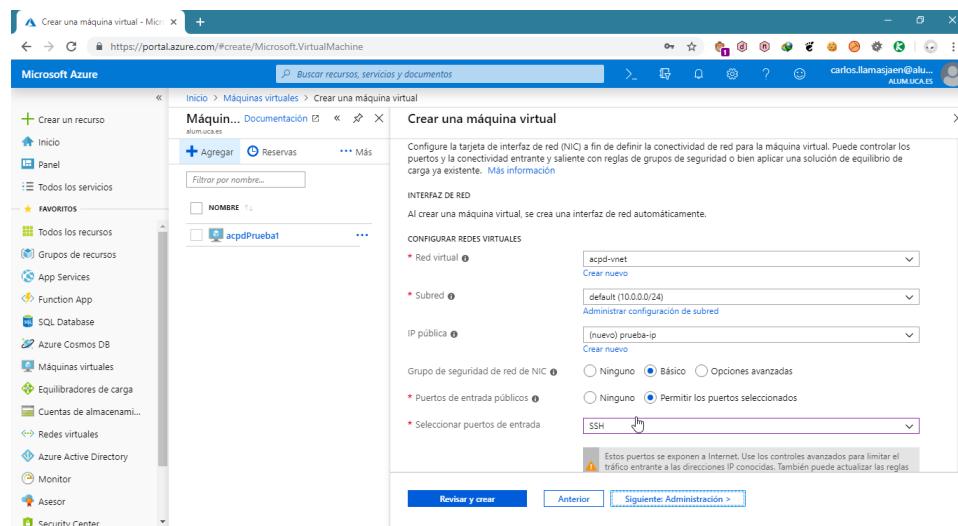


Figura 5.19: Configuración de redes virtuales.

6. Seleccionamos los puertos de entrada y si queremos equilibrio de carga:

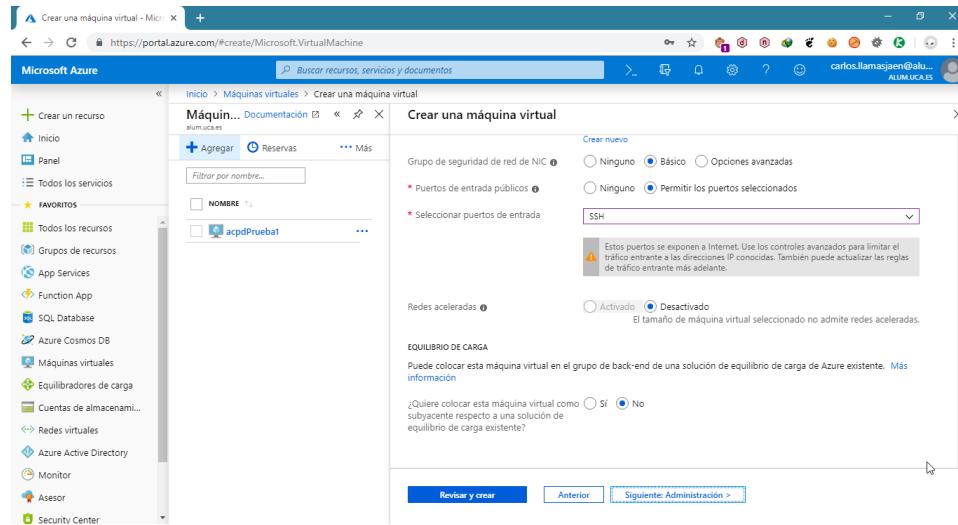


Figura 5.20: Selección de entrada y equilibrio de carga.

7. Añadimos la configuración de seguridad:

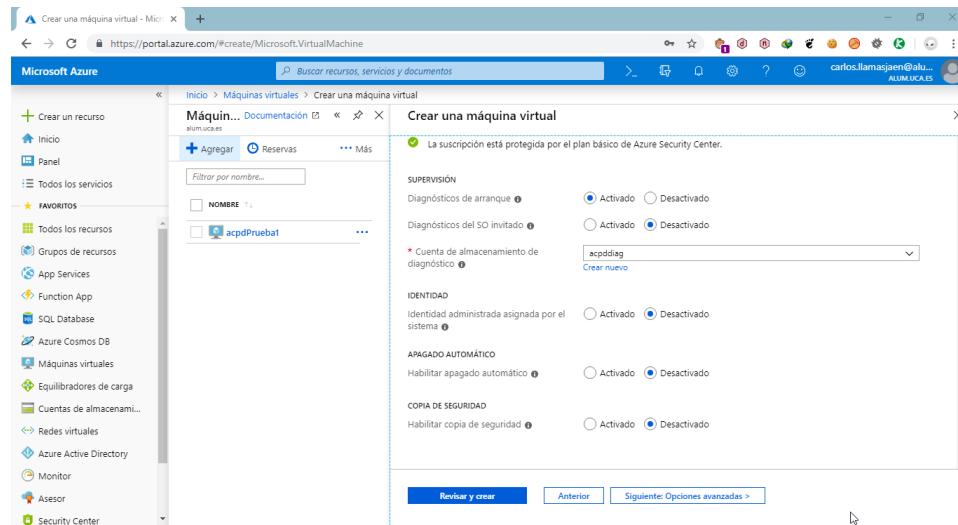


Figura 5.21: Configuración de seguridad.

8. Luego veremos las extensiones dentro de las opciones avanzadas:

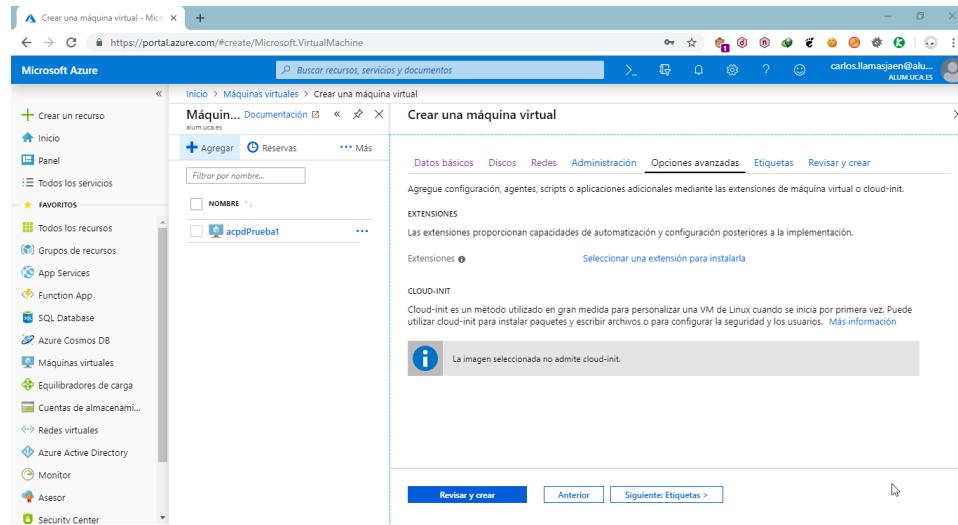


Figura 5.22: Opciones avanzadas.

9. Añadimos las etiquetas que deseemos:

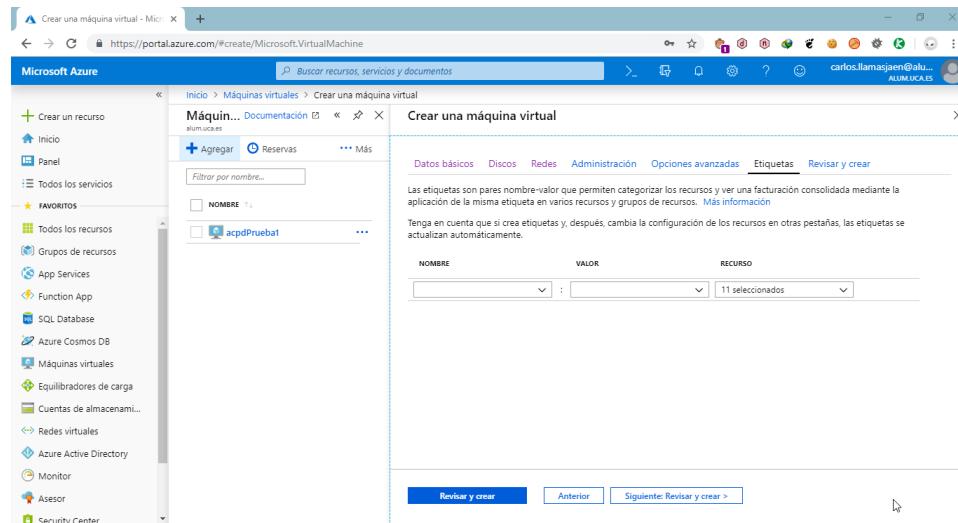


Figura 5.23: Añadimos las etiquetas.

10. Finalmente, revisamos la configuración y creamos la máquina virtual:

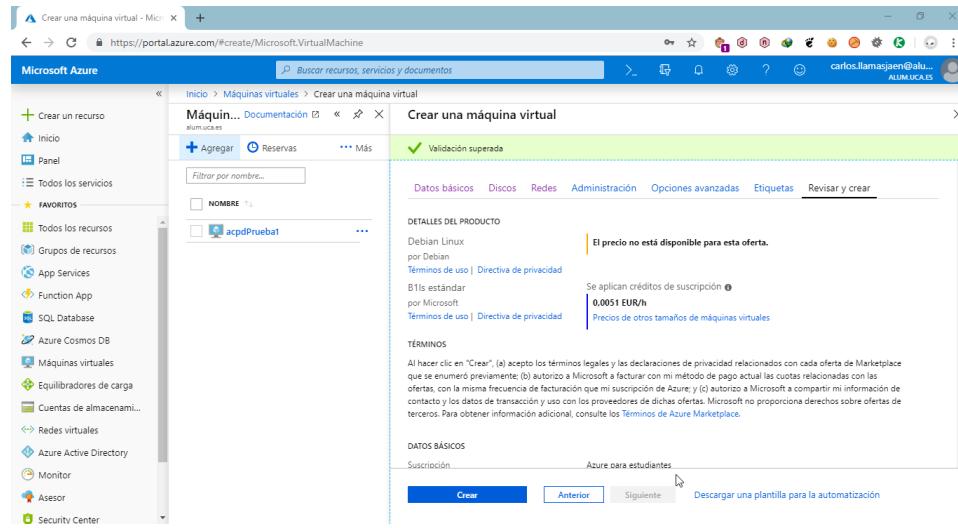


Figura 5.24: Revisión de la configuración.

11. Para conectarnos, lo haremos por ssh a la IP indicada y para ver las especificaciones de la CPU de la máquina virtual usaremos el comando cat /proc/cpuinfo:

```
carlos@prueba:~$ clear
carlos@prueba:~$ cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 79
model name     : Intel(R) Xeon(R) CPU E5-2673 v4 @ 2.30GHz
stepping        : 1
microcode      : 0xffffffff
cpu MHz        : 2294.684
cache size     : 51200 KB
physical id    : 0
siblings        : 1
core id         : 0
cpu cores      : 1
apicid          : 0
initial apicid : 0
fpu             : yes
fpu_exception   : yes
cpuid level    : 20
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 ss syscall nx pdpe1gb rdtscp lm constant_tsc

```

Figura 5.25: Información del procesador.

Capítulo 6

Creación de webs en AWS y Azure

La funcionalidad principal de estos servicios es el desarrollo de sitios webs o servicios web auto-escalables. Ambos ofrecen almacenamiento de ficheros y datos, aunque nosotros nos centraremos únicamente en la temática orientada a servicios web, ya que es el pilar de los dispositivos IoT o de los backend de las aplicaciones móviles actuales.

La ventaja de utilizar la nube frente a proveedores de hosting es la fiabilidad y el escalado, mientras que en un proveedor estándar tendríamos que realizar llamadas a servicio técnico para poder escalar, en estas plataformas los servicios web se auto-escalan en función de la carga de trabajo que deben de ejecutar. Esto incluye reducir la capacidad en casos en los que el servicio no requiera tantos recursos y así no tener que ceñirnos a ningún plan.

6.1. Servicios Web

Los servicios web son aplicaciones que proveen de información a una o varias aplicaciones por medio de distintos protocolos. Principalmente destacan dos SOAP y REST, veamos una pequeña introducción a estos protocolos.

6.1.1. SOAP

Simple Object Access Protocol, es un estándar que utiliza XML para transmitir la información, a su vez utiliza distintos componentes para informar de cómo se utiliza el servicio (WSDL) y de transmitir a los servidores qué servicios dispone un servidor concreto (UDDI). Excesivamente verboso, está en proceso de ser reemplazado por REST.

6.1.2. REST

Representation State Transfer, protocolo que utiliza HTTP para transferir el estado de los datos entre el servidor y el cliente. A diferencia de SOAP que hay que realizar una ruta distinta para manipular los datos, se aprovecha de los métodos HTTP para decidir qué hacer con un dato en concreto utilizando la misma ruta.

6.2. Creación de una web en AWS

AWS es una plataforma que lleva implícito en su nombre el proporcionar servicios webs (Amazon Web Services). Amazon ofrece una serie de recursos para crear sitios y servicios webs. Desgraciadamente, el plan de estudiante no dispone de acceso a la plataforma principal que permita crear un sitio web y no podemos poner un ejemplo como sí hemos podido hacer en el resto de apartados

Amazon dispone de los siguientes principales para la construcción de sitios y servicios webs:

- **Amazon EC2:** Se encarga de ejecutar el servidor web y las tareas de procesamiento.
- **Amazon S3:** Permite almacenar ficheros y ser accesibles por el servidor web.
- **Amazon RDS:** Sistema de gestión de bases de datos.

Eventos recientes		
Hora	Tipo	Detalles
21-05-2019 00:28:19 UTC+0200	INFO	Launched environment: PruebaAcpd2-env. However, there were issues during launch. See event log for details.
21-05-2019 00:28:17 UTC+0200	ERROR	Creating EIP 18.211.211.226 failed. Reason: Resource creation cancelled
21-05-2019 00:28:17 UTC+0200	ERROR	Creating Auto Scaling launch configuration failed Reason: API: autoscaling>CreateLaunchConfiguration User: arn:aws:sts::742345703652 assumed-role/vocstartsoft/user278720=carlos_llamasjaen@lum.uca.es is not authorized to perform: autoscaling>CreateLaunchConfiguration on resource: arn:aws:autoscaling:us-east-1:742345703652:launchConfiguration-*:launchConfigurationName/awseb-e-w7kskpcwp9-stack-AWSEBAutoScalingLaunchConfiguration-1TK47660QAA
21-05-2019 00:28:17 UTC+0200	ERROR	Stack named 'awseb-e-w7kskpcwp9-stack' aborted operation. Current state: 'CREATE_FAILED' Reason: The following resource(s) failed to create: [AWSEBAutoScalingLaunchConfiguration, AWSEBEIP].
21-05-2019 00:28:01 UTC+0200	INFO	Created security group named: awseb-e-w7kskpcwp9-stack-AWSEBSecurityGroup-1RMU82R0MORUF

Figura 6.1: Error al crear un entorno en Amazon Elastic Beanstalk

6.3. Creación de una web en Azure

En Azure sí que hemos podido desarrollar una sencilla web. La ventaja fundamental de Azure frente a AWS es que permite a través de Visual Studio publicar un sitio o servicio web con tal de hacer un click, lo que permite un desarrollo bastante más rápido que utilizar AWS. En Azure disponemos de los principales lenguajes para desarrollar aplicaciones o servicios web, en este ejemplo vamos a crear una simple página html que modificaremos en la presentación para demostrar la sencillez que tiene la integración con Visual Studio.

En Azure los sitios webs se denominan App Services, pese a que es posible desarrollar un sitio web, están más orientados a implementar un servicio web.

El primer paso es seleccionar el plan, aprovecharemos las horas gratuitas que disponemos. Tenemos que seleccionar el dominio, el cual podemos utilizar de los que Azure proporciona de forma gratuita.

Datos básicos Supervisión Etiquetas Revisar y crear

App Service Web Apps le permite generar, implementar y escalar rápidamente aplicaciones empresariales web, móviles y de API que se ejecutan en cualquier plataforma. Satisface los estrictos requisitos de rendimiento, escalabilidad, seguridad y cumplimiento sin renunciar a una plataforma totalmente administrada para el mantenimiento de la infraestructura. [Más información](#)

DETALLES DEL PROYECTO

Seleccione una suscripción para administrar los recursos implementados y los costos. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

* Suscripción: Azure para estudiantes
└─ * Grupo de recursos: acpd [Crear nuevo](#)

DETALLES DE INSTANCIA

* Nombre: prueba-acpd [.azurewebsites.net](#)

* Publicar: Código [Imagen de Docker](#)

* Pila del entorno en tiempo de ejecución: .NET Core 2.2

* Sistema operativo: Windows [Linux](#)

* Ubicación: Central US

PLAN DE APP SERVICE

El plan de tarifa de App Service determina la ubicación, las características, los costos y los recursos del proceso asociados a la aplicación. [Más información](#)

* Plan: (Nuevo) ASP-acpd-9747 [Crear nuevo](#)

Figura 6.2: Creación del sitio web

Una vez seleccionado el plan se nos pide seleccionar si queremos métricas, que hemos desactivado ya que son de pago. En el siguiente paso se nos pide si queremos asignar etiquetas de facturación al recurso.

Inicio > App Services > Aplicación Web (versión preliminar)

Aplicación Web (versión preliminar)

Crear

¿Busca la experiencia de creación clásica de Web Apps? →

Datos básicos Supervisión Etiquetas Revisar y crear

Azure Monitor le proporciona visibilidad completa de las aplicaciones, la infraestructura y la red. [Más información](#)

APPLICATION INSIGHTS

Habilitar Application Insights [No](#) [Sí](#)

Figura 6.3: Selección de métricas

Datos básicos Supervisión Etiquetas Revisar y crear

Las etiquetas son pares nombre-valor que permiten categorizar los recursos y ver una facturación consolidada mediante la aplicación de la misma etiqueta en varios recursos y grupos de recursos.

Tenga en cuenta que si crea etiquetas y, después, cambia la configuración de los recursos en otras pestañas, las etiquetas se actualizan automáticamente.

NOMBRE	VALOR	RECURSO
<input type="text"/>	<input type="text"/> :	<input type="text"/> Aplicación web

Figura 6.4: Selección de etiquetas

Una vez hecho esto tenemos el sitio ya arrancado, pese a que muestra una página de demostración.

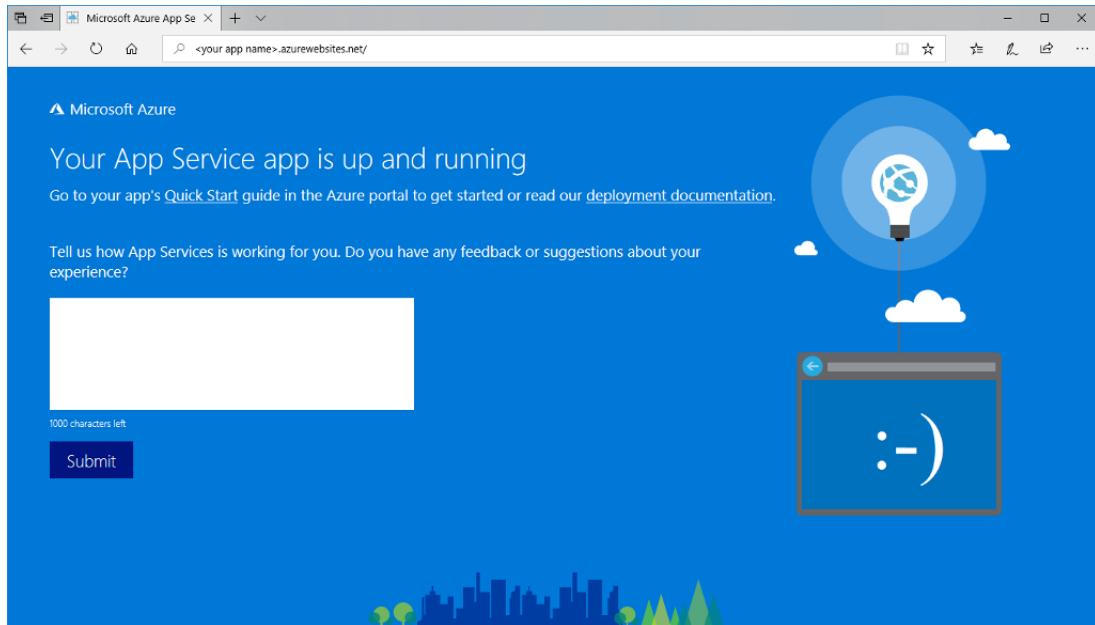


Figura 6.5: Página de demostración

Ahora pasaremos a subir una página desde Visual Studio. Para ello es necesario descargar el perfil de publicación, para poder subir el sitio al servidor.

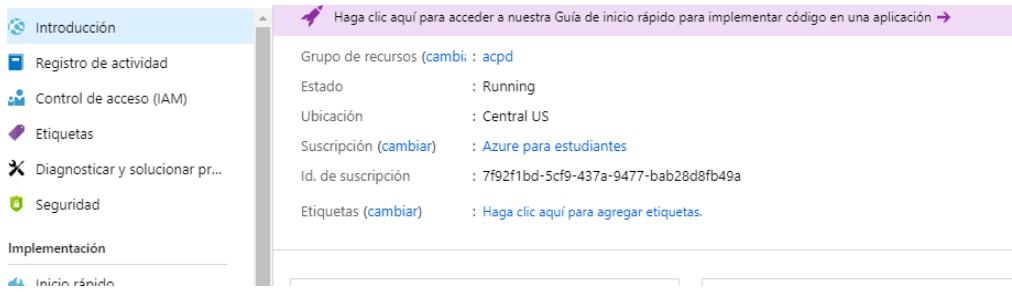


Figura 6.6: Centro de control del sitio web

El código de la web es muy simple, únicamente imprime un código HTML con una foto y un texto.

```
namespace PruebaACPD
{
    public class Startup
    {
        // This method gets called by the runtime. Use this method to add services to the container.
        // For more information on how to configure your application, visit https://go.microsoft.com/fwlink/?linkid=398948
        public void ConfigureServices(IServiceCollection services)
        {

        }

        // This method gets called by the runtime. Use this method to configure the HTTP request pipeline.
        public void Configure(IApplicationBuilder app, IWebHostEnvironment env)
        {
            if (env.IsDevelopment())
            {
                app.UseDeveloperExceptionPage();
            }

            app.Run(async context =>
            {
                await context.Response.WriteAsync("<html><body></img></body></html>");
            });
        }
    }
}
```

Figura 6.7: Código fuente

Finalmente seleccionamos publicar en el explorador de soluciones y cargamos el perfil de publicación. Solamente tenemos que hacer click en publicar, para que el sitio se suba a Azure.



Figura 6.8: Publicación de sitio web

```
Conectándose a ftp://waws-prod-dm1-139.ftp.azurewebsites.windows.net/site/wwwroot...
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\MSBuild\Current\Bin\Roslyn\csc.exe /noconfig /unsafe- /checked-
PruebaACPD -> C:\Users\Carlos\source\repos\PruebaACPD\PruebaACPD\bin\Release\netcoreapp2.1\PruebaACPD.dll
PruebaACPD -> C:\Users\Carlos\source\repos\PruebaACPD\PruebaACPD\obj\Release\netcoreapp2.1\PubTmp\Out\
Publicando carpeta ...
La aplicación web se publicó correctamente ftp://waws-prod-dm1-139.ftp.azurewebsites.windows.net/site/wwwroot
La aplicación web se publicó correctamente http://prueba-acpd.azurewebsites.net/
```

Figura 6.9: Publicación correcta

Cargamos el sitio web y podemos ver que ha cambiado



Prueba ACPD

Figura 6.10: Web una vez subida

Con esto ya tendríamos listo el sitio web, obviamente este es un ejemplo sencillo, un ejemplo más complejo lo tenemos en la página web smallpdf.com, es una página web que realiza diversas tareas de manipulación sobre ficheros PDF y que utiliza la nube de AWS para procesar y almacenar temporalmente los archivos PDF ya manipulados.

Capítulo 7

Creación de servicios de IoT en AWS y Azure

Tanto Azure como AWS ofrecen servicios de mensajería entre dispositivos, como por ejemplo colas de mensajes.

Ambos utilizan MQTT para la comunicación, aunque es posible en Azure utilizar su propio protocolo, pero no ofrece ventaja alguna sobre MQTT.

Vamos a demostrar las características de creación y administración de colas de mensajes y los dispositivos que se conectan a ella.

Las ventajas de utilizar un servicio para este tipo de proyectos es el ahorro en costes de infraestructura y de seguridad que se tendría que realizar de forma manual, ya que habría que disponer de un servidor dedicado, exponerlo con seguridad a internet. Sin embargo, utilizando estos servicios nos tenemos que preocupar únicamente de desarrollar las aplicaciones, tampoco suponen un coste desorbitado, ya que para pocos dispositivos (50) es gratuito y a partir de ahí, el millón de mensajes entre dispositivos varía entre 0,7 y 1,5 dólares.

Además cabe mencionar la escalabilidad, que es la gran estrella de los servicios en la nube, poder escalar en caso de necesidad sin suponer una gran inversión ni tiempo.

7.1. Creación de servicios IoT en AWS

En AWS tenemos que crear primero un IoT Center. AWS es sencillo de conectar, ya que al revés que Azure, en el cuál predomina su propio protocolo, se basa en el protocolo MQTT para transmitir los datos.

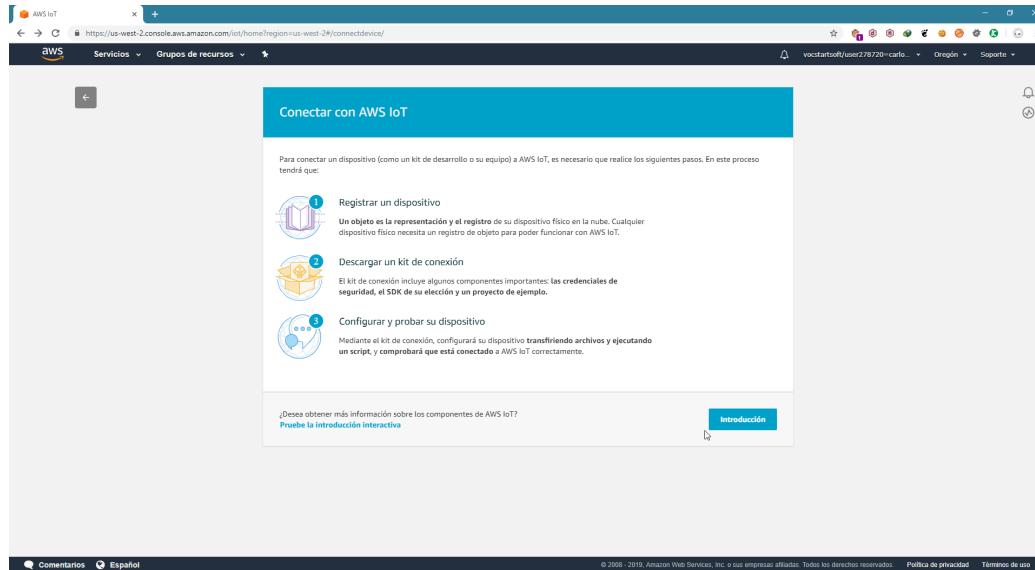


Figura 7.1: Creación del centro IOT

Tras ver la introducción, se nos pide seleccionar el lenguaje a utilizar y el sistema operativo, para así poder descargar las herramientas que nos permitirán empezar a hablarnos con AWS a una cola MQTT.

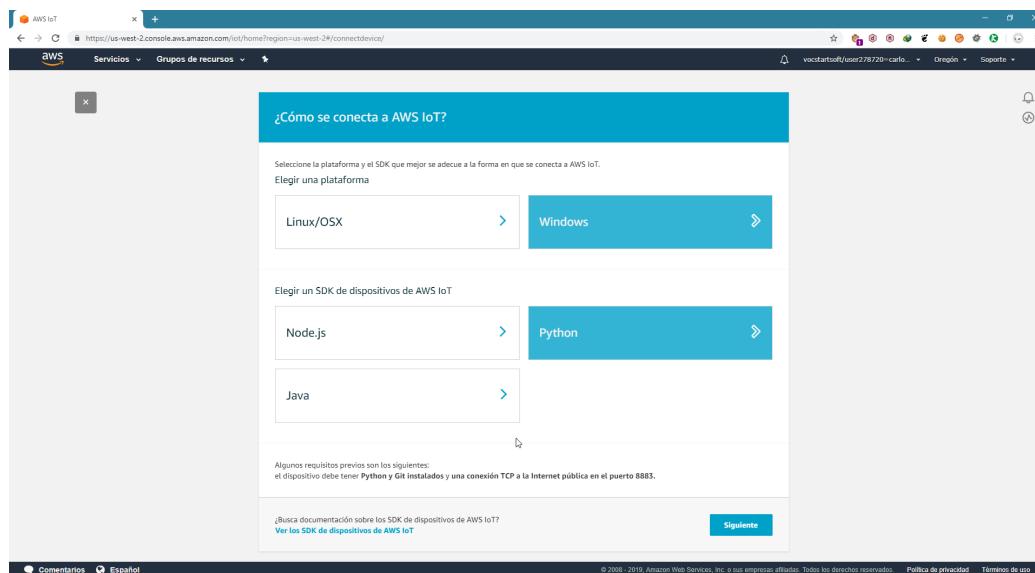


Figura 7.2: Selección del sistema operativo

Posteriormente tenemos que asignarle un nombre al objeto / dispositivo para poder reconocerlo en las métricas, ya que la autenticación, como se verá más adelante se realiza por medio de certificados.

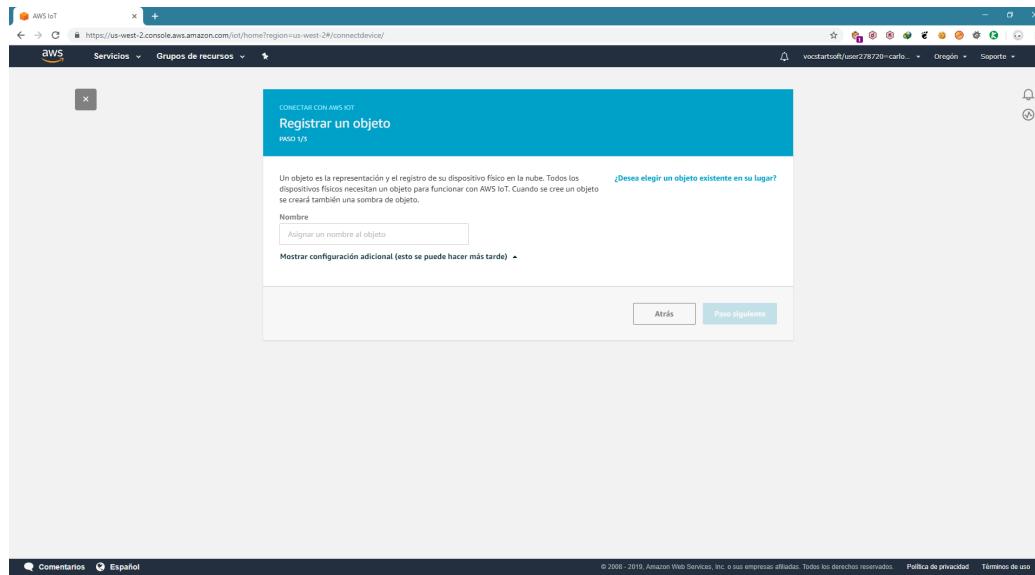


Figura 7.3: Selección del nombre del dispositivo

Una vez creado el nombre, estamos listos para ejecutar los ejemplos que vienen con el kit de desarrollo, viene con todo lo necesario para empezar a comunicarnos con AWS, tanto los ejemplos como los certificados necesarios para autenticar el objeto anterior.

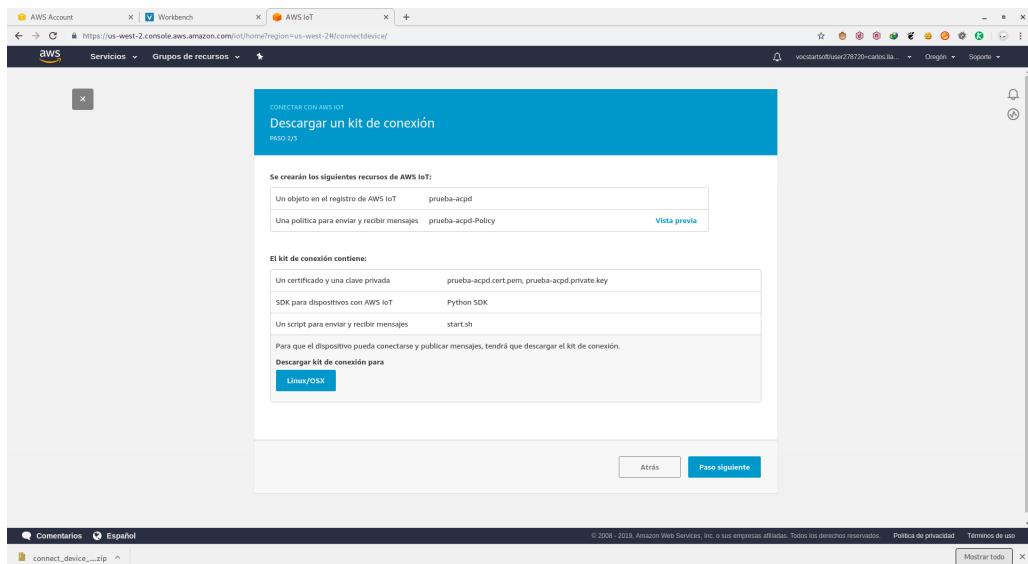


Figura 7.4: Descarga del kit de prueba

Posteriormente lo ejecutamos y empezaremos a ver como llegan mensajes y la aplicación de Amazon es capaz de leerla.

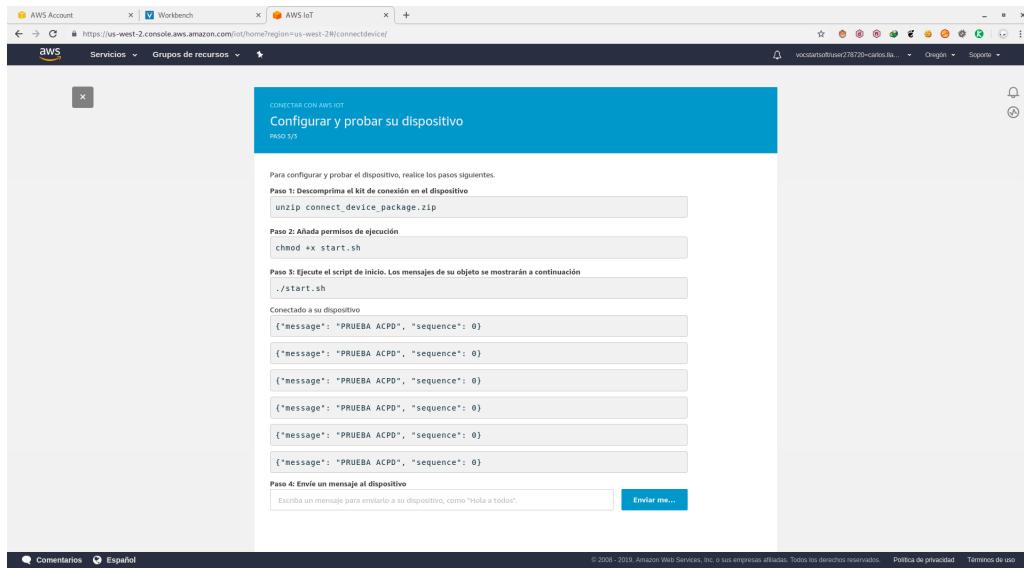


Figura 7.5: Prueba de ejecución

Una vez llegado a este paso tenemos todo lo necesario para empezar a desarrollar nuestras aplicaciones sobre IoT.

También resulta interesante, añadir nuevos dispositivos / objetos, visualizar métricas de mensajes, modificar las políticas de seguridad de los dispositivos.

Para ello disponemos en el centro de IoT la sección de métricas:

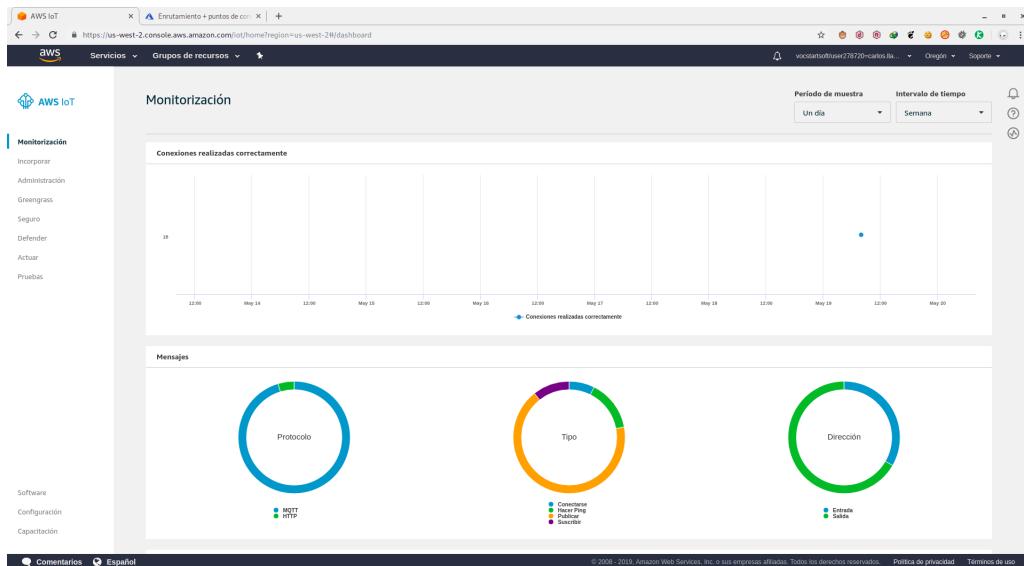


Figura 7.6: Visualización de las métricas

También disponemos una interfaz de gestión de dispositivos, donde podemos gestionar los parámetros de estos dispositivos:

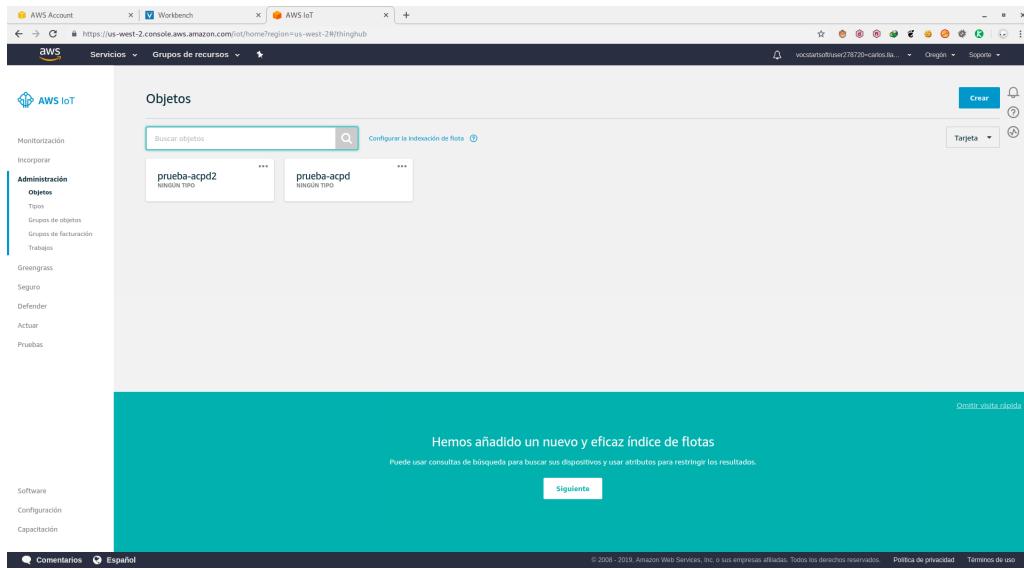


Figura 7.7: Visualización del número de dispositivos

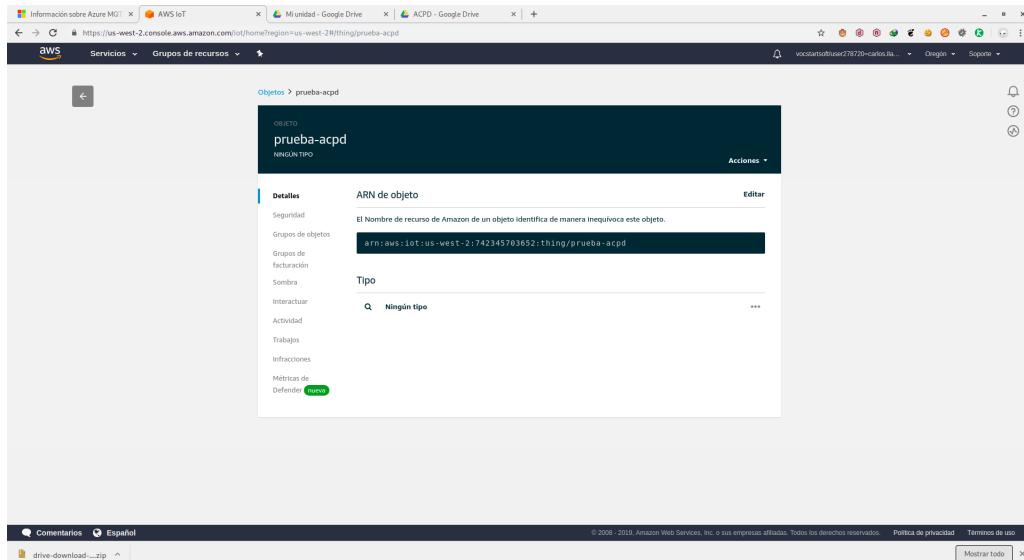


Figura 7.8: Gestión de un dispositivo

7.2. Creación de servicios IoT en Azure

Ahora vamos a centrarnos en el servicio de Microsoft, la ventaja que tiene sobre AWS es la multiplataforma y la dedicación de Microsoft al IoT, ya que tiene distintas versiones de Windows adaptadas al IoT. Mientras que AWS soporta únicamente Java, Node.JS y Python, Azure lo expande con su plataforma estrella, .NET Core, a la que le está dando un gran empujón en los últimos años.

La ventaja de poder usar .NET Framework o .NET Core para desarrollar estas aplicaciones es la integración con el IDE

Visual Studio, ya que prácticamente con 1 click podemos enviar la aplicación en la nube, como se vió en el apartado de desarrollo de servicios web.

Lo primero que hay que realizar en Azure es dar de alta un Centro de IoT o IoT Hub. Los grupos de recursos de Azure van enfocados a agrupar los recursos de cara a la facturación, en este caso al disponer de crédito gratuito podemos ignorarlo, con cuidado de no sobrepasar los límites.

The screenshot shows the 'Centro de IoT' creation wizard. The 'Aspectos básicos' tab is selected. The form includes fields for Subscription (Azure para estudiantes), Resource Group (acpd), Region (Oeste de EE. UU.), and IoT Hub Name (prueba-acpd). A note at the top says: 'Cree una instancia de IoT Hub que le ayude a conectar, supervisar y administrar miles de millones de recursos de IoT. [Obtener más información](#)'.

Figura 7.9: Creación de un centro IoT

Lo siguiente que se nos pide es la escala, a diferencia de AWS que no necesita escalar, si no que cobra por tramos de mensajes, Azure es necesario predefinir una escala, aunque no lleguemos a ella siempre se nos cobrará lo mismo. En este caso Azure tiene una versión gratuita para 8000 mensajes diarios.

The screenshot shows the 'Escala y tamaño' tab of the IoT Hub configuration. It's set to 'F1: Nivel Gratis'. Below it, a slider for 'Número de unidades de IoT Hub de F1' is set to 1. A note says: 'Esto determina la funcionalidad de escalabilidad de IoT Hub y se puede cambiar a medida que aumente la necesidad.' A detailed table below lists features for the F1 tier: Mensajes del dispositivo a la nube (Habilitado), Mensajes al día (8.000), Enrutamiento de mensajes (Habilitado), Costo al mes (0.00 EUR), Comandos de nube a dispositivo (Habilitado), IoT Edge (Habilitado), and Administración de dispositivos (Habilitado).

Figura 7.10: Selección de la escala

Lo próximo que hay que realizar es añadir un dispositivo nuevo. A diferencia de AWS, Azure sí que te deja poner una clave en vez que utilizar certificados.

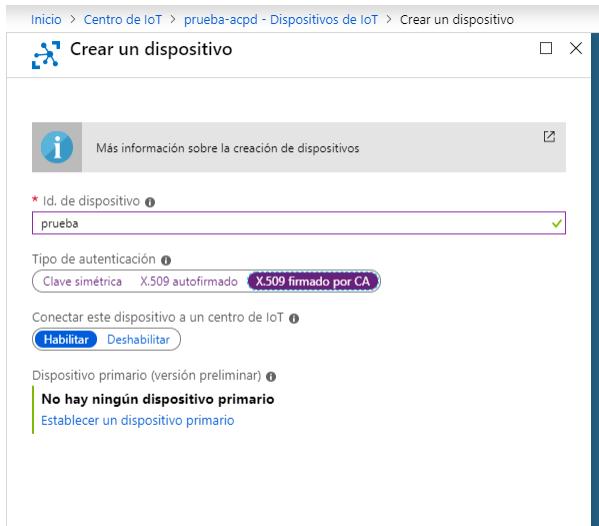


Figura 7.11: Creación de dispositivo

Podemos ver los dispositivos en la lista de dispositivos.

Figura 7.12: Lista de dispositivo

Una vez listo podemos volver al centro de IOT. Desde aquí podemos establecer políticas de seguridad, filtros, gestionar certificados, mirar métricas, eventos y gestionar la mensajería.

Figura 7.13: Centro de IoT

Pues con todo esto listo vamos a probar a ejecutar un ejemplo, para ello tenemos que crear otro dispositivo más y obtener las claves, es posible que haya que utilizar la CLI de Azure, que es de pago, ya que tiene que crear una pequeña cuenta de almacenamiento para poderla usar.

```
Bash
carlos@Azure:~$ az iot hub show --query properties.eventHubEndpoints.events.endpoint --name prueba-acpd
Comprehensive IoT data-plane functionality is available in the Azure IoT CLI Extension.
To install the extension, run: "az extension add --name azure-cli-iot-ext"
For more info and install guide go to: https://github.com/Azure/azure-iot-cli-extension

carlos@Azure:~$ az extension add --name azure-cli-iot-ext
carlos@Azure:~$ az iot hub show --query properties.eventHubEndpoints.events.endpoint --name prueba-acpd
"sb://[REDACTED].servicebus.windows.net/"
carlos@Azure:~$ az iot hub show --query properties.eventHubEndpoints.events.path --name prueba-acpd
"iothub[REDACTED]"
carlos@Azure:~$
```

Figura 7.14: Utilización de la Azure CLI para obtener las claves de los dispositivos.

Una vez establecidas las claves, probamos un ejemplo que hay de IoT Hub, este ejemplo un publicador envía la temperatura de una sala y un suscriptor la recibe. En rojo aparece marcados los mensajes correspondientes que se envían desde el publicador y aparece en el suscriptor.

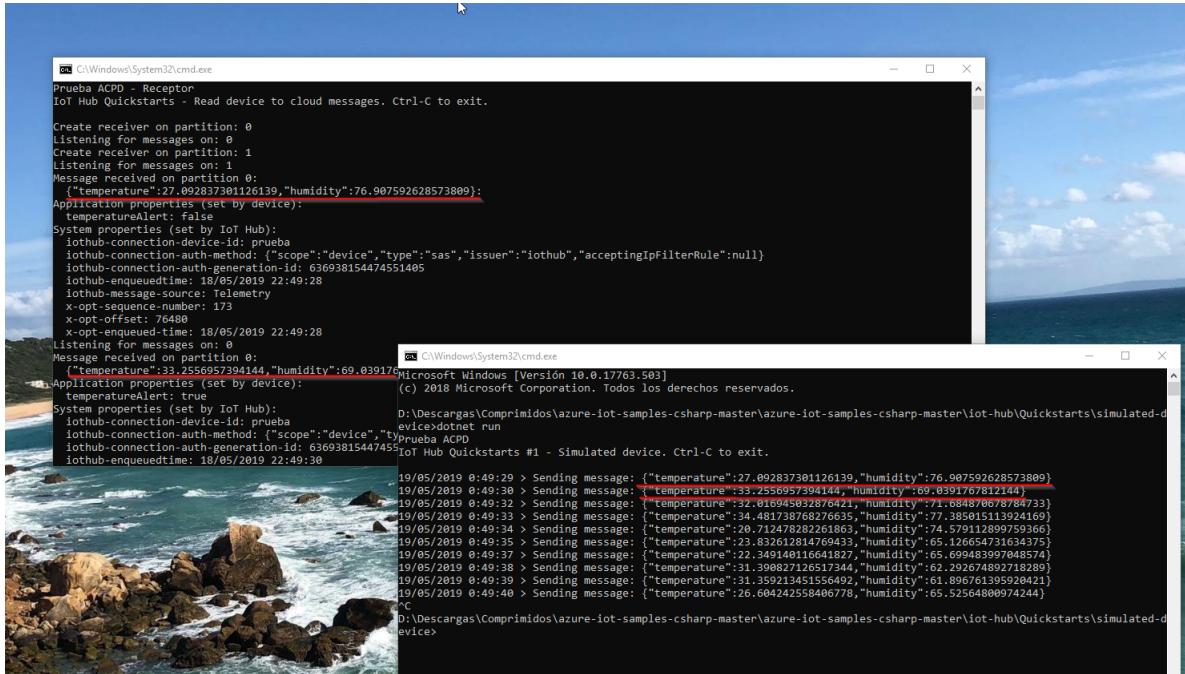


Figura 7.15: Ejemplo de IoT

Parte IV

Anexo

Capítulo 8

Referencias

En este capítulo se detallarán las referencias consultadas a la hora de redactar este documento:

- Documentación oficial de AWS: <https://aws.amazon.com/es/>.
- Documentación oficial de Azure: <https://azure.microsoft.com/es-es/>.
- https://es.wikipedia.org/wiki/Amazon_Web_Services.
- <https://es.wikipedia.org/wiki/Middleware>.
- <https://revistadigital.inesem.es/informatica-y-tics/cloud-computing-con-amazon/>.
- https://docs.aws.amazon.com/es_es/general/latest/gr/aws_service_limits.html.
- <https://azure.microsoft.com/es-es/free/free-account-students-faq/>.
- https://es.wikipedia.org/wiki/Microsoft_Azure
- <https://docs.microsoft.com/es-es/azure/iot-hub/quickstart-send-telemetry-dotnet>