



Tema 8. Despliegue de aplicaciones web

Programación web

Boni García
Curso 2017/2018

Índice

1. Introducción
2. Infraestructura propia
3. Servicios de alojamiento
4. Servicios en la nube

Índice

1. **Introducción**
2. Infraestructura propia
3. Servicios de alojamiento
4. Servicios en la nube

Introducción

- En esta asignatura hemos aprendido como desarrollar aplicaciones y servicios web con diferentes tecnologías:
 - Lado cliente: HTML, CSS, JavaScript, jQuery, Bootstrap, Angular
 - Lado servidor: Spring Boot (MVC, Thymeleaf, Security, JPA)
 - Servicios web: Spring MVC
 - Aplicaciones móviles híbridas: Ionic
- En este tema vamos a aprender como **desplegar** nuestras aplicaciones web en entornos de producción

Introducción

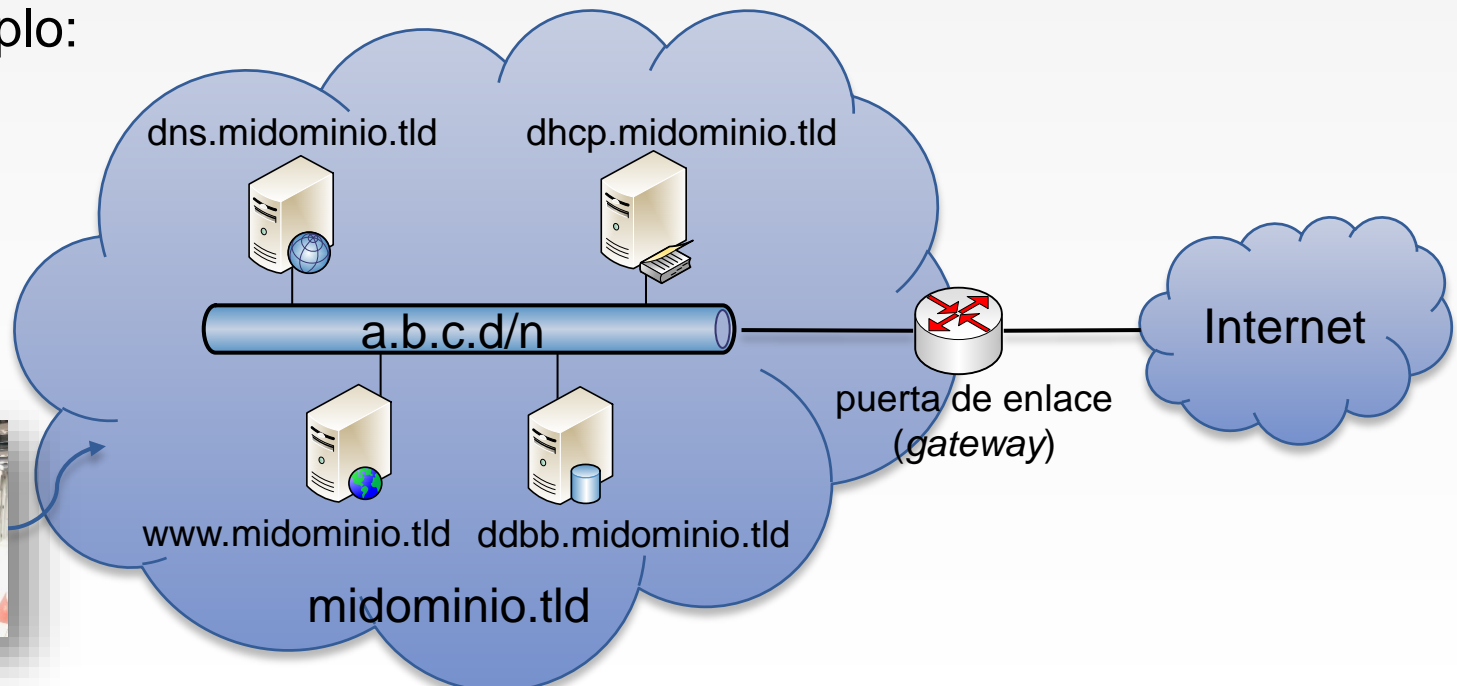
- En la actualidad hay tres enfoques principales para el despliegue de aplicaciones y servicios web:
 1. Usando infraestructura propia (*on premises*)
 2. Mediante servicios de alojamiento (*hosting*)
 3. Mediante servicios en la nube (*cloud computing*)

Índice

1. Introducción
2. **Infraestructura propia**
3. Servicios de alojamiento
4. Servicios en la nube

Infraestructura propia

- Para desplegar una aplicación web necesitamos un **servidor web** en una máquina con **IP pública** y típicamente con un **nombre dominio** asociado (esto es, dado de alta un servidor DNS autoritativo)
- Por ejemplo:



Infraestructura propia

- Cuando lidiamos directamente con la infraestructura, el despliegue de aplicaciones está supeditado al tipo de servidor (y base de datos)
- En servidores de aplicaciones Java, lo habitual consiste en desplegar aplicaciones empaquetadas como:
 - WAR: *Web Application Archive*
 - EAR: *Enterprise Application Archive*
- En las aplicaciones que hemos visto en clase (con Spring-Boot) el servidor Tomcat viene embebido, con lo que el despliegue de la aplicación simplemente consiste en ejecutar la aplicación principal (empaquetada como JAR)



```
java -jar my-app.jar
```



Índice

1. Introducción
2. Infraestructura propia
3. **Servicios de alojamiento**
4. Servicios en la nube

Servicios de alojamiento

- Los servicios de alojamiento (*hosting*) proporcionan un espacio dedicado a sus clientes para el despliegue de aplicaciones web
- Típicamente, además proporcionan servicios de DNS, correo electrónico, base de datos, copia de seguridad (*backup*), y diferentes tecnologías de servidor (Java, PHP, .NET, ...)
- Algunos de los más conocidos en España:



<http://www.1and1.es/>



<https://www.strato.es/>



<http://www.arsys.es>

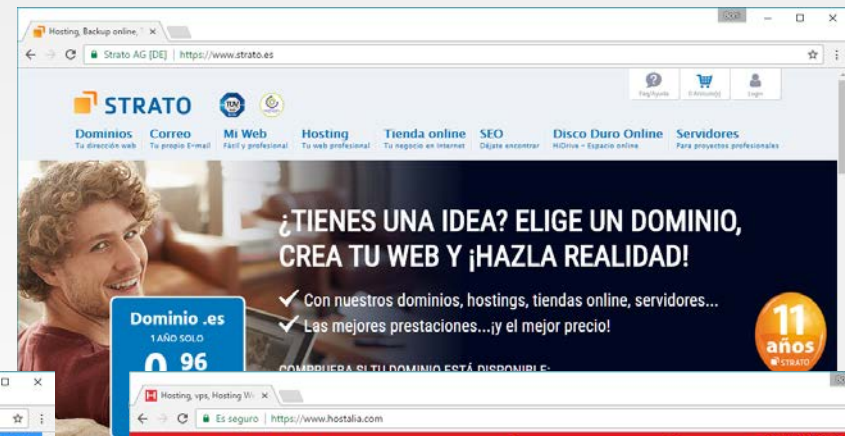
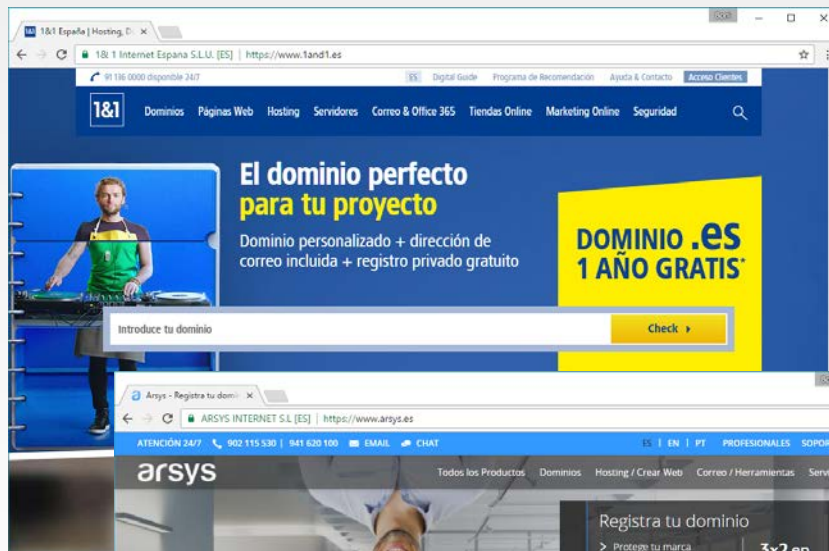


<http://www.hostalia.com>



<http://www.sync.es>

Servicios de alojamiento



Índice

1. Introducción
2. Infraestructura propia
3. Servicios de alojamiento
4. Servicios en la nube
 - *Cloud computing*
 - *Software as a Service (SaaS)*
 - *Platform as a Service (PaaS)*
 - *Infrastructure as a Service (IaaS)*
 - Heroku
 - Amazon Web Services

Servicios en la nube

Cloud computing

- La computación en la nube (*cloud computing*) es el término con el se conoce al modelo que permite el **acceso remoto** a un conjunto de **recursos** (redes, servidores, aplicaciones, almacenamiento, y servicios) que pueden ser rápidamente provisionados con un **mínimo esfuerzo** de gestión y configuración



<https://www.nist.gov/programs-projects/cloud-computing>

Servicios en la nube

Cloud computing

- Las características fundamentales de los servicios en la nube son:
 - **Acceso remoto:** Los servicios son distribuidos (accesible a través de Internet)
 - **Elasticidad:** ampliación o reducción de los recursos de la plataforma en función de las necesidades puntuales
 - **Recursos compartidos:** los recursos (pueden ser físicos, aunque típicamente serán virtualizados) pueden ser compartidos por diferentes usuarios
- Normalmente los servicios en la nube son accesibles a través de:
 - Aplicaciones web
 - Servicios REST

Servicios en la nube

Cloud computing

- Existen 3 tipos de servicios *cloud* fundamentales:

1. **Software as a Service** (SaaS): Software como servicio

- Se puede ver como un modelo de distribución de software a través de Internet (soluciones en la nube de alto nivel)

2. **Platform as a Service** (PaaS): Plataforma como servicio

- Modelo de servicio que ofrece una plataforma y las herramientas de programación para el despliegue de aplicaciones y servicios web (soluciones en la nube nivel medio)

3. **Infrastructure as a Service** (IaaS): Infraestructura como servicio

- Recursos, físicos y virtuales (soluciones en la nube de bajo nivel)

Servicios en la nube

Cloud computing

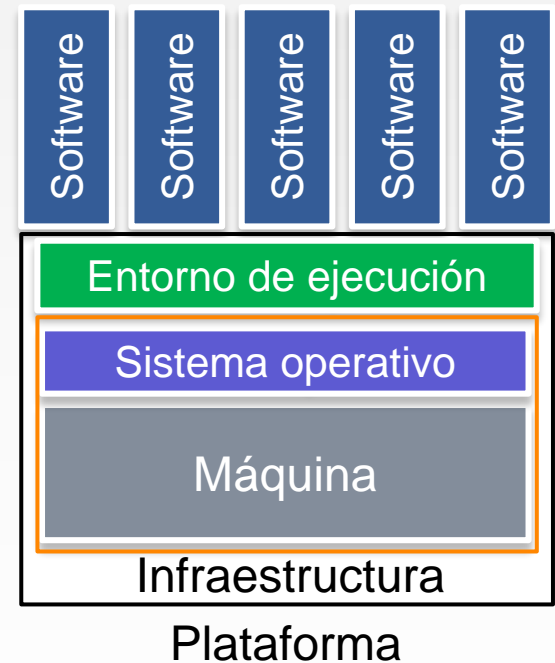
- Vamos a recuperar conceptos ya conocidos para intentar asimilar los conceptos de SaaS, PaaS, e IaaS
- En primer lugar conviene recordar lo que entendemos por **servicio distribuido**

Un **servicio distribuido** consiste en varios procesos que se ejecutan en diferentes equipos terminales y que se comunican a través de una red de datos (típicamente Internet)

Servicios en la nube

Cloud computing

- **Software** = programas que se ejecutan en una determinada plataforma
- **Plataforma** = Infraestructura en la que opcionalmente se puede encontrar un entorno de ejecución (por ejemplo Java o Node.js)
- **Infraestructura** = máquina (física o virtual) en la cual existe un sistema operativo instalado para poder interactuar con el hardware



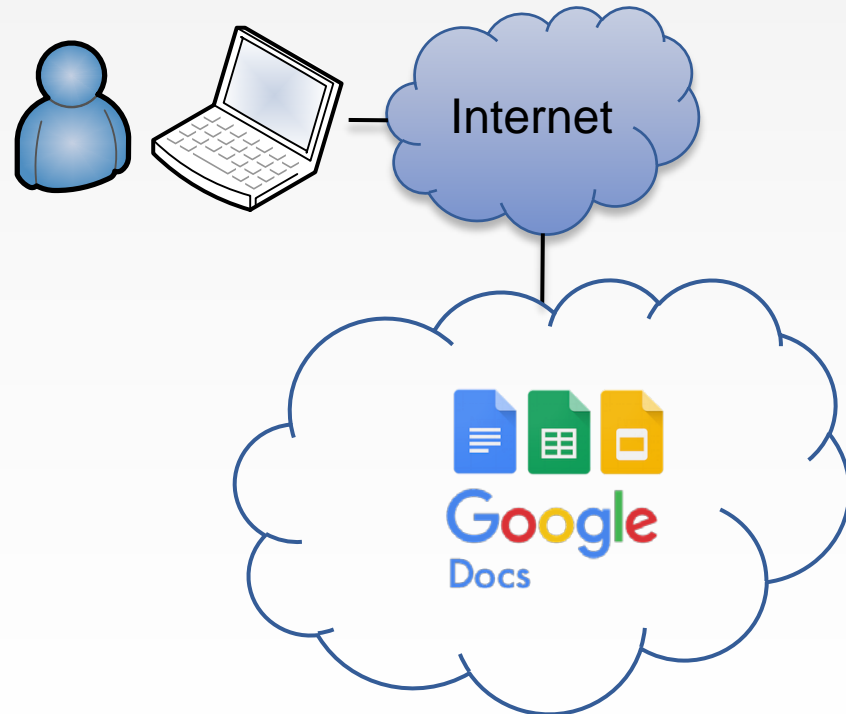
Servicios en la nube

Cloud computing

- Software



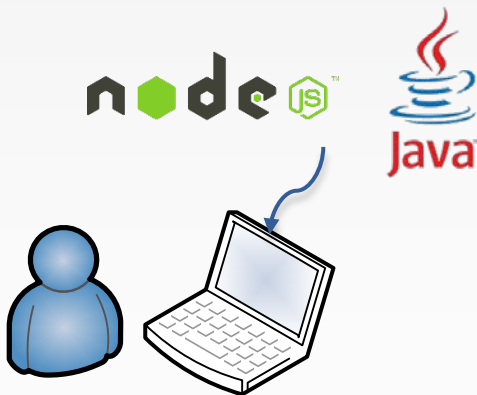
- *Software as a Service (SaaS)*



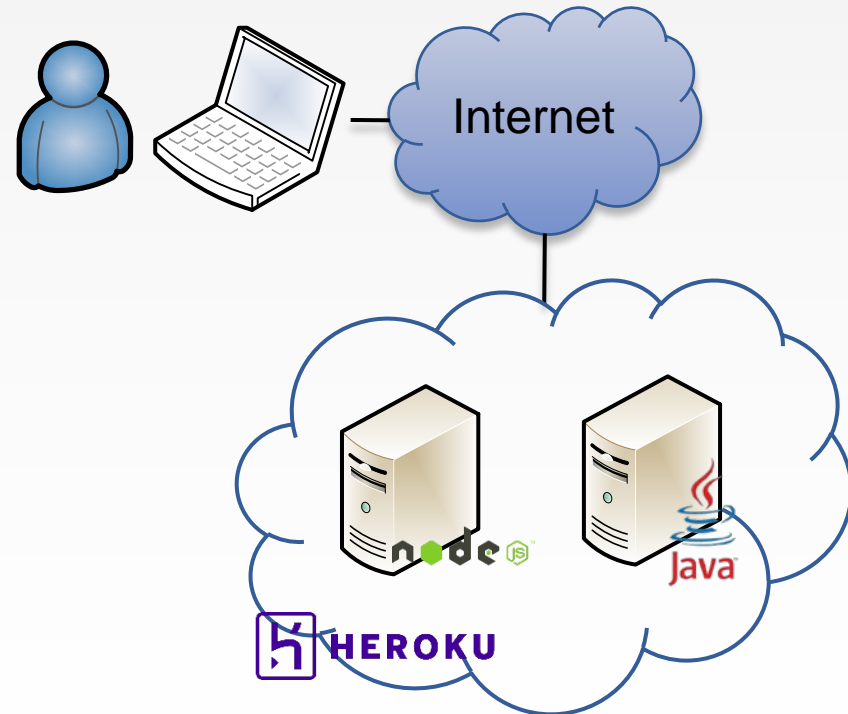
Servicios en la nube

Cloud computing

- Plataforma



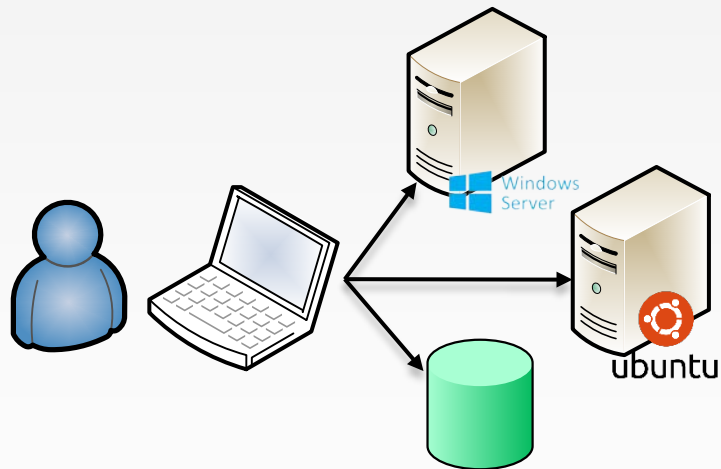
- *Platform as a Service (PaaS)*



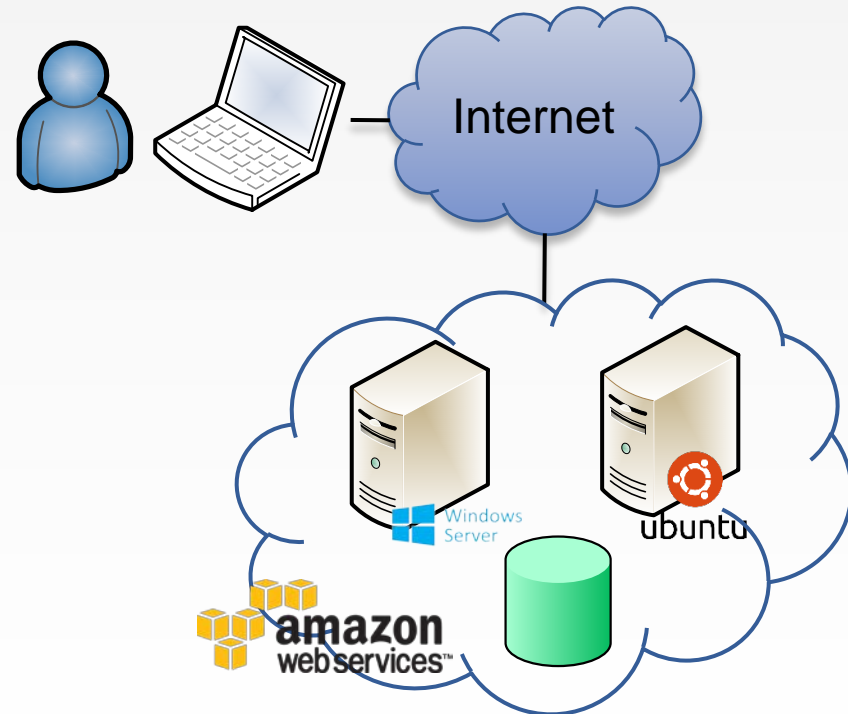
Servicios en la nube

Cloud computing

■ Infraestructura



■ *Infrastructure as a Service (IaaS)*



Servicios en la nube

Cloud computing

- Hoy día ya se habla de XaaS (*Anything as a Service*)
- En general, cualquier servicio ofrecido como solución *cloud* puede encajar en este modelo genérico. Por ejemplo:



Jenkins



Travis CI

Continuous Integration (CI) server

CI as a Service



Firebase

Mobile Backend as a Service (BaaS)

Servicios en la nube

Software as a Service (SaaS)

- El software como servicio engloba aquellos servicios que se ofrecen al **usuario final** vía web (también típicamente mediante REST)
- Desde un punto de vista técnico, servicios de PaaS (incluso IaaS) se podrían considerar como SaaS, pero con SaaS se incide en que el usuario final usa el servicio
- Si es de pago, el modelo suele ser por uso o por número de usuarios
- Ejemplos: Google Apps, Microsoft 365, Dropbox, iCloud



Servicios en la nube

Plataforma como servicio

- En el **Platform as a Service (PaaS)** se ofrece una plataforma para la ejecución de aplicaciones (servidores web, bases de datos, gestión de logs, monitorización, auto-escalado, etc...)
- Los desarrolladores no se preocupan de la gestión de la plataforma, sólo se preocupan del desarrollo de su software
- Existen diversos **proveedores de PaaS** como por ejemplo: Heroku, Appfog, OpenShift, CloudFoundry, Google App Engine, etc



Servicios en la nube

Infraestructura como servicio

- Es la capa de abstracción más baja del *cloud computing*
- Servicios típicos ofrecidos por un proveedor IaaS:
 - Servidores (*instances*)
 - Balanceadores de carga (*load balancer*)
 - Gestión de sistemas operativos (*images*)
 - Copias de seguridad
 - Almacenamiento de datos
 - Direcciones IP
- Los proveedores IaaS más utilizados son Amazon Web Services (AWS) y Microsoft Azure

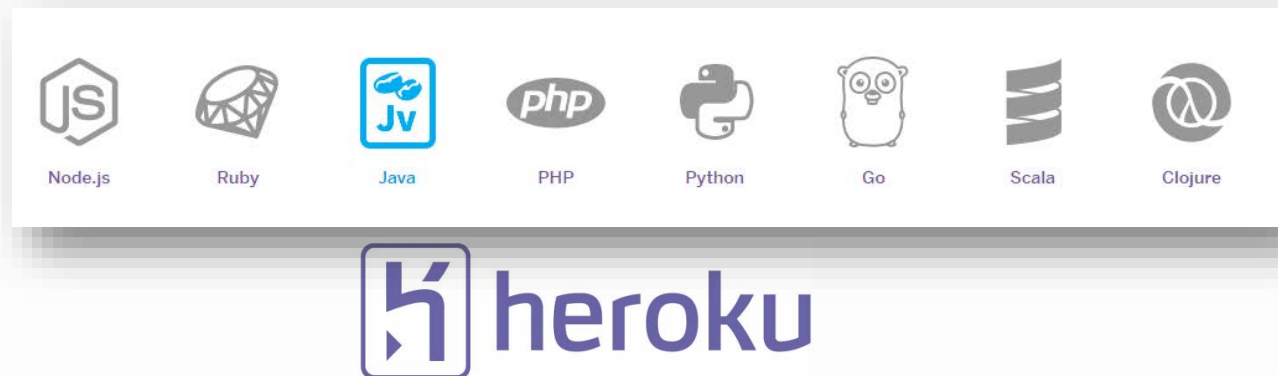
Tanto Azure como AWS también ofrecen servicios PaaS



Servicios en la nube

Heroku

- **Heroku** es un servicio PaaS propiedad de la empresa Salesforce
- Permite desplegar aplicaciones implementadas con diferentes tecnologías: Node.js, Ruby, Java, PHP, Python, Go, Scala, Clojure
- Ofrece base de datos PostgreSQL, MongoDB, Redis, Cloudant, Couchbase Server



<https://www.heroku.com/>

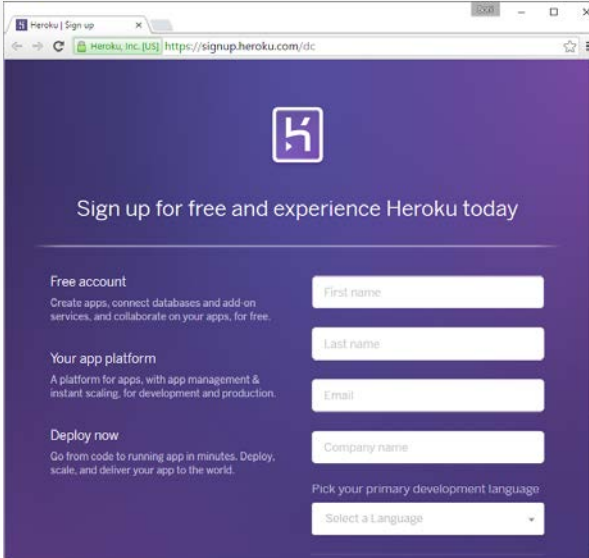
<https://devcenter.heroku.com/>

Servicios en la nube

Heroku

- Vamos a ver cómo desplegar una aplicación **Angular** en Heroku
- El primer paso será crearse una cuenta gratuita en Heroku:

<https://signup.heroku.com/dc>



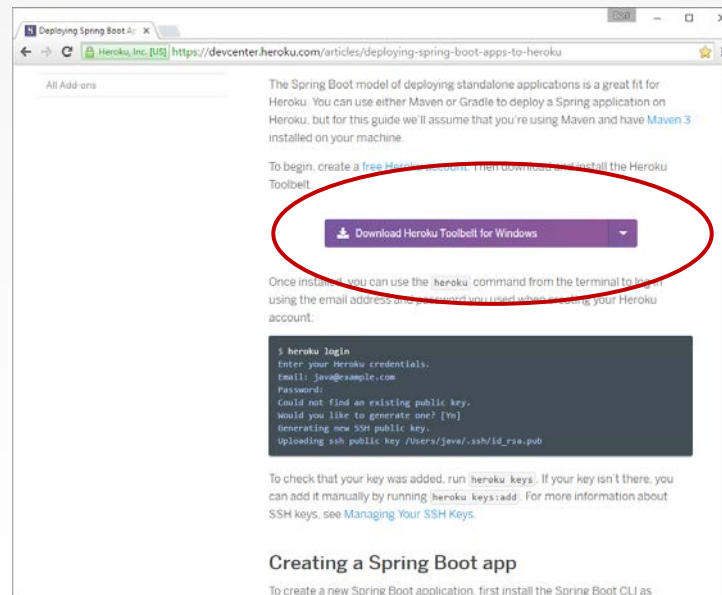
The screenshot shows the Heroku sign-up page in a web browser. The page has a purple header with the Heroku logo and the text "Sign up for free and experience Heroku today". Below this, there are three sections: "Free account", "Your app platform", and "Deploy now". Each section has a brief description and a corresponding form field. The "Free account" section has a "First name" field. The "Your app platform" section has "Last name" and "Email" fields. The "Deploy now" section has a "Company name" field and a "Pick your primary development language" dropdown menu. The browser's address bar shows the URL "https://signup.heroku.com/dc".

Servicios en la nube

Heroku

- Después habrá que instalar la herramienta de Heroku para línea de comandos (Heroku *Command Line Interface*, CLI)

<https://devcenter.heroku.com/articles/deploying-spring-boot-apps-to-heroku>



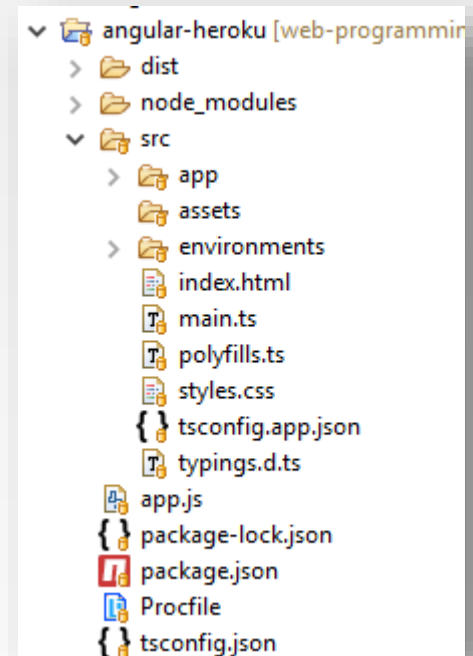
Servicios en la nube

Heroku

- En el directorio donde tenemos nuestra aplicación Angular hay que crear un fichero de configuración para Heroku llamado Procfile
- Este fichero le dice a Heroku como hay que lanzar nuestra aplicación dentro del PaaS

Procfile

```
web: node app.js
```



Fork me on GitHub

Servicios en la nube

Heroku

- Hay que crear el fichero `app.js`:

```
var express = require('express');
var http = require('http');
var path = require('path');

var app = express();

app.use(express.static(path.join(__dirname, 'dist')));

app.get('*', function(req, res) {
  res.sendFile(path.join(__dirname, 'dist/index.html'));
});

const port = process.env.PORT || '3001';
app.set('port', port);

const server = http.createServer(app);
server.listen(port, () => console.log('Running'));
```

Servicios en la nube

Heroku

- Hay que modificar el fichero `package.json`:

```
{
  ...
  "scripts": {
    ...
    "postinstall": "ng build --aot -prod"
  },
  "engines": {
    "node": "~6.11.0",
    "npm": "~3.10.10"
  },
  "dependencies": {
    "@angular/cli": "~1.7.4",
    "@angular/compiler-cli": "^5.2.0",
    ...
  },
  "devDependencies": {
    "@angular/language-service": "^5.2.0",
    "typescript": "~2.5.3"
  }
}
```

Servicios en la nube

Heroku

- Hay que preparar nuestra aplicación para usar Git:

```
git init  
git add .  
git commit -m "first commit"
```

- Después creamos nuestra aplicación con el siguiente comando:

```
heroku create
```

- Para subir los cambios mediante Git:

```
git push heroku master
```

- Para ver nuestra aplicación:

```
heroku open
```

- Para ver la salida de log de nuestra aplicación:

```
heroku logs --tail
```

En este punto Heroku CLI nos pedirá credenciales (login/password) de la cuenta que hemos creado previamente

Servicios en la nube

Amazon Web Services

- **Amazon Web Services** (AWS) es el proveedor de servicios *cloud* más usado en la actualidad
- AWS ofrece un amplio conjunto de servicios IaaS y PaaS
- Soporta diferentes modelos de pago
 - Bajo demanda
 - Instancias reservadas
 - Instancias de subastas (pujar por capacidad libre en AWS)



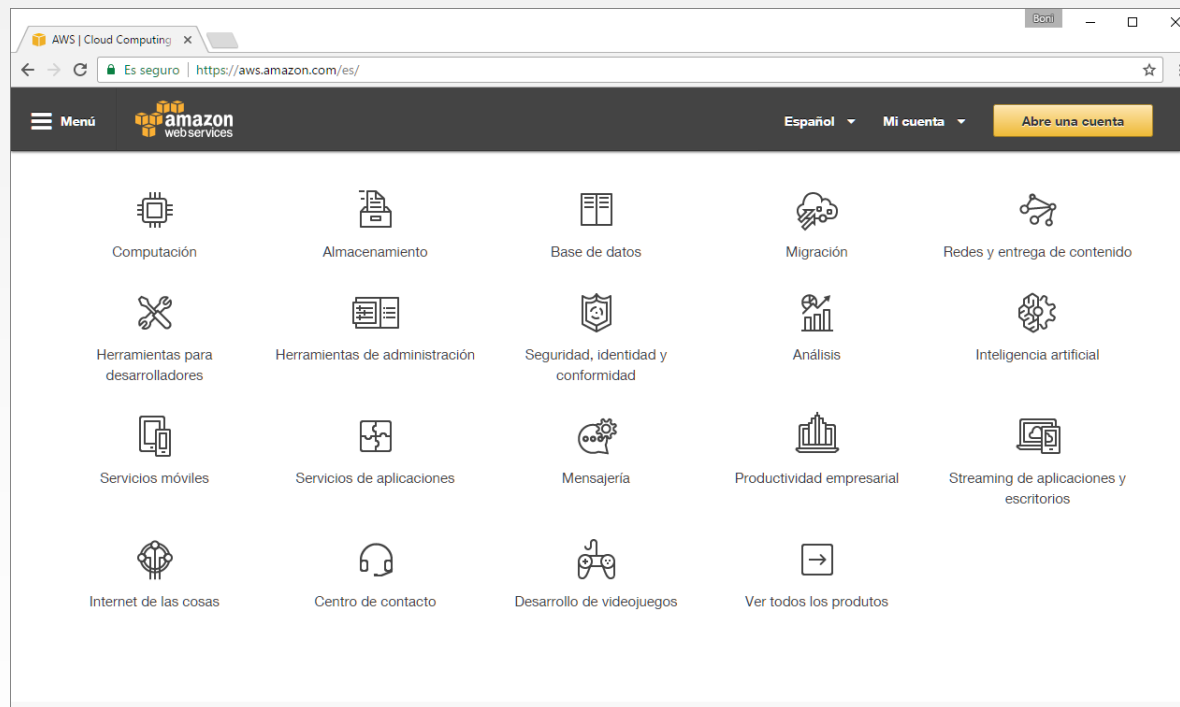
Servicios en la nube

Amazon Web Services

- Algunos de los principales servicios AWS son:
 - Servidores virtuales: **Amazon EC2**
 - Almacenamiento escalable: **Amazon S3**
 - Bases de datos: **Amazon RDS** (bases de datos relacionales), **Amazon DynamoDB** (bases de datos no relacionales)
 - Redes privadas: **Amazon VPC**
 - Monitorización de recursos y aplicaciones: **Amazon CloudWatch**
 - ...

Servicios en la nube

Amazon Web Services



<https://aws.amazon.com/>

Servicios en la nube

Amazon Web Services

- Los diferentes servicios AWS se ofrecen en diferentes regiones geográficas
 - EEUU Oeste, EEUU Este, Canadá, América del sur, Europa, Asia pacífico, China



<https://aws.amazon.com/es/about-aws/global-infrastructure/regional-product-services/>

Servicios en la nube

Amazon Web Services

- *Amazon Elastic Compute Cloud* (**Amazon EC2**) es un servicio *cloud* que proporciona servidores virtuales (o instancias) “en la nube” (IaaS)
- Disponen de varios tipos de instancias según su “hardware”
- Hay diferentes tipos de tamaño para cada tipo de instancia:
 - Nano, micro, small, medium, large, large, xlarge, 2xlarge, 4xlarge, 10xlarge, 16xlarge

<https://aws.amazon.com/es/ec2/instance-types/>

Servicios en la nube

Amazon Web Services

Familia de instancias EC2

Familia	Tipo	Características
Uso general	T2	Procesadores Intel Xeon de alta frecuencia (CPU en ráfagas)
Uso general de última generación	M3	Equilibrio de recursos informáticos, de memoria y red
	M4	Similar a M3 con mejoras en las funciones de red
Optimizadas para uso general	C3	Procesadores Intel Xeon E5-2666 optimizados para EC2
	C4	Similar a C3 con almacenamiento elástico (<i>Elastic block store</i> , EBS)
Optimizadas para memoria	X1	Ofrecen el costo más bajo por GiB de RAM
	R4	Recomendadas para uso intensivo de memoria (DDR4)
	R3	Costo más bajo por GiB que R4
Instancias aceleradas	P2	Destinadas a aplicaciones de informática GPU de uso general
Optimizadas para uso de gráficos	G2	Streaming de aplicaciones 3D, codificación de vídeo, etc
Hardware personalizable	F1	Basadas en tecnología FPGU (<i>Field Programmable Gate Array</i>)
Optimizadas para almacenamiento	I3	Optimizado para baja latencia de E/S

Servicios en la nube

Amazon Web Services

- Los precios en de las instancias EC2 dependen de
 - De la región
 - Del tipo de instancia
 - Del momento del día (oferta/demanda)

<https://aws.amazon.com/es/ec2/pricing/on-demand/>

- AWS nos ofrece una calculadora para poder estimar precios

<https://calculator.s3.amazonaws.com/index.html>

Servicios en la nube

Amazon Web Services

- Hay diferentes formas de pago para instancias EC2:
 - Instancias bajo demanda (*on-demand*):
 - El usuario paga exclusivamente por el uso que hace de las instancias (se paga por horas)
 - Instancias reservadas:
 - El usuario reserva instancias pagando una cantidad por adelantado para 1 o 3 años
 - En las instancias reservadas el coste por hora es menor
 - Subasta de instancias (*spot instances*):
 - El usuario fija el precio al que está dispuesto a pagar una hora de cómputo
 - Amazon ofrece al usuario las instancias “que sobran” cuando están disponibles a ese precio en base a la oferta y la demanda

Servicios en la nube

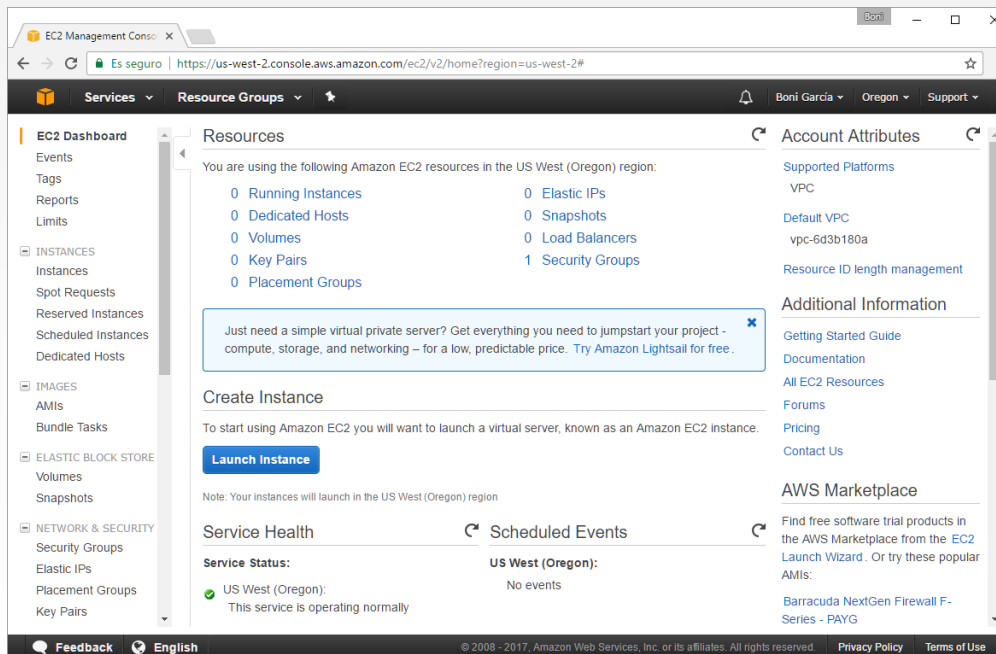
Amazon Web Services

- El servicio de EC2 incluye también:
 - Imágenes (*Amazon Machine Image*, AMI)
 - Existe un mercado de AMIs: <https://aws.amazon.com/marketplace/>
 - Grupos de seguridad
 - Grupos de autoescalado
 - Monitorización
 - Balanceadores de carga

Servicios en la nube

Amazon Web Services

■ Ejemplo: creación de una instancia EC2



<https://console.aws.amazon.com/>

Servicios en la nube

Amazon Web Services

■ Ejemplo: creación de una instancia EC2

The screenshot shows the AWS Management Console interface for the EC2 Launch Wizard. The first step, 'Choose an Amazon Machine Image (AMI)', is active. It displays two AMIs: 'Red Hat Enterprise Linux 7.4 (HVM), SSD Volume Type' and 'Ubuntu Server 16.04 LTS (HVM), SSD Volume Type'. Both are marked as 'Free tier eligible'. A 'Select' button is next to each. At the bottom, there is a promotional banner for Amazon RDS.

The screenshot shows the second step, 'Choose an Instance Type'. It provides information about Amazon EC2 instance types and a table of available options. The 'Currently selected' instance is 't2.micro'.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GiB)	EBS-Optimized Available	Network Performance	IP Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Y
<input checked="" type="checkbox"/>	General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Y
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Y
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Y
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Y

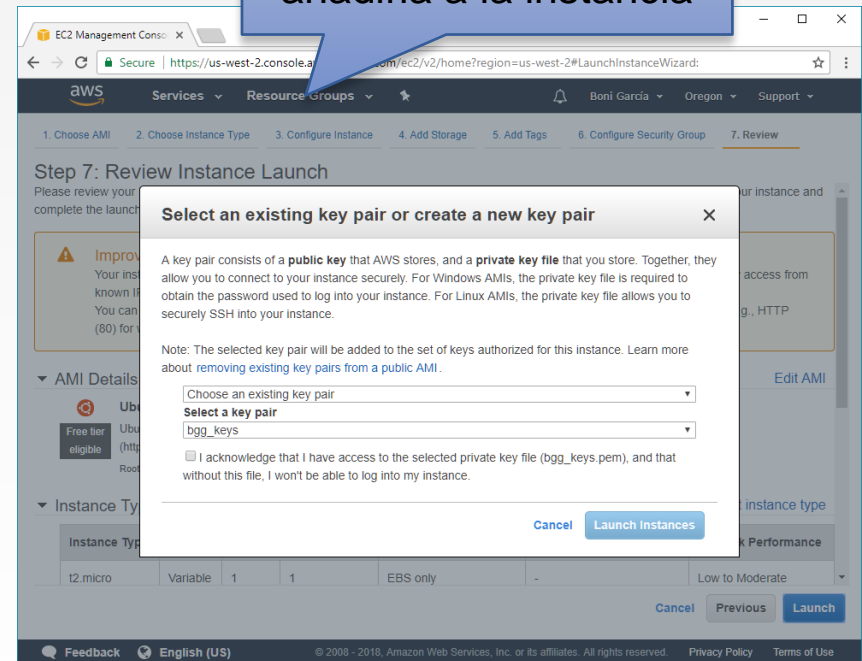
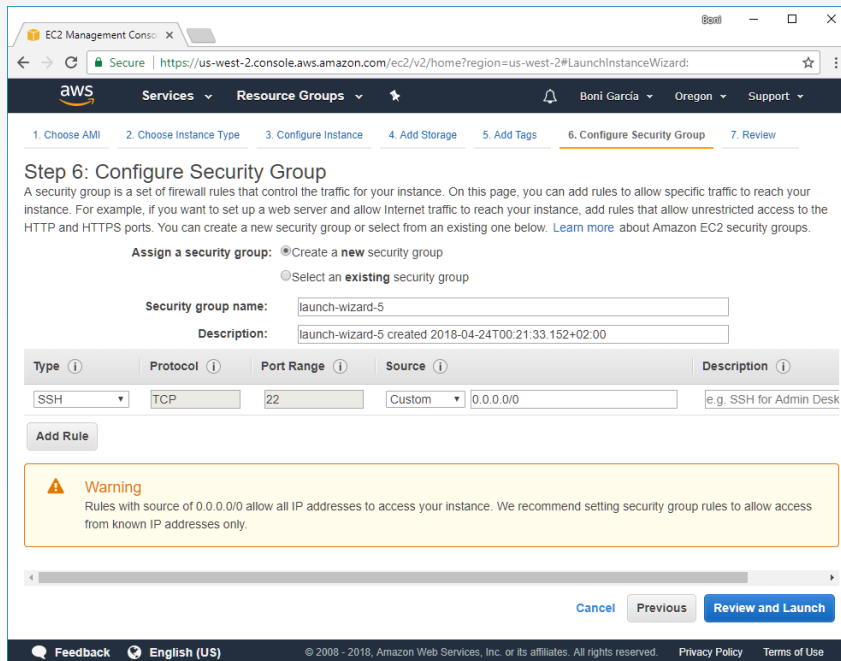
Buttons at the bottom: Cancel, Previous, Review and Launch, Next: Configure Instance Details.

Servicios en la nube

Amazon Web Services

■ Ejemplo: creación de una instancia EC2

Necesitaremos un par de claves SSH. La clave pública habrá que añadirla a la instancia



Servicios en la nube

Amazon Web Services

■ Ejemplo: creación de una instancia EC2

```
> ssh -i .ssh/id_rsa ec2-54-245-152-115.us-west-2.compute.amazonaws.com
```

```
> sudo nohup java -jar -Dserver.port=80 spring-boot-hello-world-1.0.0.jar
```

Nos podemos conectar a la instancia mediante SSH

Podemos arrancar una aplicación spring-boot desde la línea de comandos

Hay que asegurarse que el grupo de seguridad tiene el puerto que hayamos elegido para el servidor web abierto hacia el exterior

