Curso de Back-End con Node.js (Avanzado)

Clase 09

Temario

Temario

- Deployment.
- Ejercicios.

Deployment

“Terminamos de desarrollar nuestro proyecto,

¿ahora cómo lo publicamos?”

Software Deployment (1/3)

Son todas las actividades que se deben llevar a cabo para poner el software a disposición del usuario (para que lo pueda usar).

Si se va poner el software a disposición del usuario final, se habla de *“Deploy to Production”*.

En las aplicaciones web, esto implica colocar el software en un servidor público o privado. En general se usan servidores públicos, pero a veces hay aplicaciones que sólo están disponibles en la red de una organización (ej: Intranet).

Software Deployment (2/3)

La tarea de configurar un servidor suele ser algo complicado y generalmente está en manos de un SysAdmin o un DevOps; no en las de un desarrollador.

Por ejemplo, hay que tener en cuenta temas como:

- Instalación y configuración de un servidor (ej: Apache o Nginx).
- Instalación y configuración de una base de datos (ej: MySQL, MongoDB).
- Instalación de extensiones (como las que se requieren en PHP).
- Instalación y configuración de un firewall.
- Configuración de los permisos de los archivos.
- Gestión de backups.
- Instalación y configuración de un balanceador de cargas.
- Gestión de actualizaciones.
- Etc, etc, etc.

Pero si les interesa el tema, pueden vichar tutorial: https://serversforhackers.com/s/start-here.

Software Deployment (3/3)

Una de las primeras tareas a realizar es elegir un hosting para la aplicación.

Se podría contratar una máquina virtual en AWS EC2 o DigitalOcean pero, si bien son excelentes opciones, requieren de ciertos conocimientos que escapan a los objetivos de este curso.

Por suerte, existen servicios como Heroku, Vercel y AWS Elastic Beanstalk que simplifican mucho la tarea de deployment. Son servicios que tienen como premisa que el programador se enfoque en su código y no la infraestructura.

Nota: Recientemente DigitalOcean lanzó un producto llamado App Platform, que está en línea con Heroku y Vercel.

Deployment @ Vercel

Vercel (1/6)

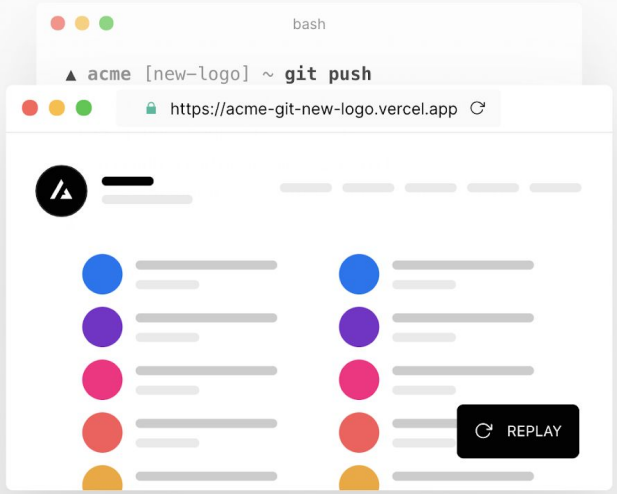
Develop. Preview. Ship.

Vercel is the optimal workflow for frontend teams.
All-in-one: Static and Jamstack deployment,
Serverless Functions, and Global CDN.

Deploy Free

Contact Sales

Takes 15 seconds



Vercel (2/6)

Vercel (antes llamado Zeit) es un servicio de hosting que permite que el desarrollador se concentre en el desarrollo de las apps y no “perder tiempo” en tareas de SysAdmin o DevOps. Es similar a opciones como Heroku y Netlify.

Algunas de sus características son:

- Está construído “arriba” de AWS Lambda.
- Es serverless → el desarrollador no debe preocuparse de la infraestructura y sólo se paga por el tiempo real de uso.
- Tiene un servicio gratuito excelente, que puede ser usado en producción (no sólo para *tests*).

Vercel (3/6)

Vercel permite importar el código de una aplicación presente en un repositorio Git. Para ello será necesario crearse una cuenta en Vercel (crear un proyecto) y especificar la URL del repositorio Git que se utilizará. Además, Vercel permite *automatic deploys*, por lo que cada vez que haya una modificación en el repositorio, automáticamente se hará el deploy en Vercel.

Vercel (4/6)

Otras características de Vercel:

- Incluye HTTPS por defecto.
- No permite alojar archivos subidos por los usuarios en el *file system*.
- No incluye base de datos ni cloud storage (hay que conseguirlo a parte).
- Permite crear URLs como: https://mi-app.vercel.app.
- Permite configurar dominios propios (ej: miempresa.com).
- Es necesario configurar las variables de entorno (que normalmente se colocan en un archivo `.env`) desde el *dashboard* de Vercel.

OPCIONAL

Vercel (5/6)

Si bien lo ideal suele ser linkear Vercel con nuestra cuenta de **GitHub** (y configurar *automatic deploys*), también se puede “deployar” la aplicación con un simple comando de consola:

```
vercel
```

Para hacer uso del mismo hay que previamente instalar Vercel CLI de forma global:

```
npm i -g vercel
```

Ver detalles.

Vercel (6/6)

A la hora de “deployar” una aplicación Node.js/Express en Vercel, es necesario crear un archivo llamado `vercel.json` en la carpeta raíz de la aplicación:

```
{
  "version": 2,
  "builds": [
    { "src": "server.js", "use": "@vercel/node" },
  ],
  "routes": [
    {
      "src": "/(.*)",
      "dest": "/server.js"
    }
  ]
}
```

Con este JSON le indicamos a Vercel cómo debe ejecutar los archivos que componen la aplicación. Ver detalles.

MongoDB Atlas

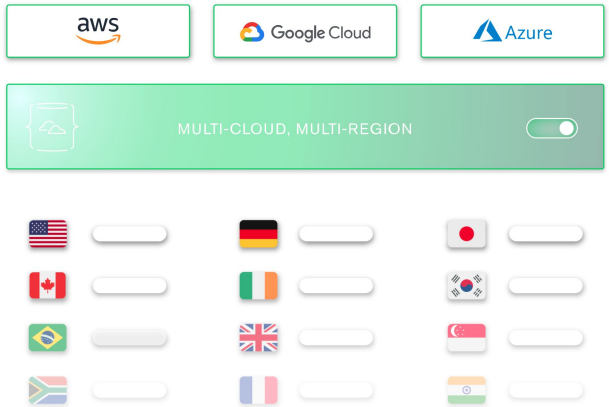
Alojamiento para bases de datos MongoDB

MongoDB Atlas (1/4)

Dado que Vercel no brinda almacenamiento para bases de datos, es necesario buscar otro proveedor.

MongoDB Atlas es un servicio brindado por MongoDB Inc, la empresa creadora de la base de datos MongoDB. Es lo que se conoce como *database-as-a-service*.

Dispone de una versión gratuita que brinda 512 MB de almacenamiento sin costo.



MongoDB Atlas (2/4)

Instructivo:

1. Crear una cuenta gratuita en MongoDB Atlas.
2. Si en algún momento se les pregunta por el *“Cloud Provider”*, seleccionar AWS (*Amazon Web Services*).
3. Al crear una cuenta, se creará (de forma automática) una “Organización” con el nombre de ustedes. Ej: *“María's Org”*. Al principio, la organización tendrá un único miembro (ustedes mismos). Si lo desean podrían agregar miembros adicionales a dicha organización. También podrán crear nuevas organizaciones. Por ejemplo: *“Equipo 5 Org”*.

MongoDB Atlas (3/4)

Instructivo (cont):

4. Dentro de las organizaciones, existirá uno o más “Proyectos”. Esto sirve para poder organizar las bases de datos con mayor granularidad, si es que tienen muchas BD. Al principio, seguramente tendrán un único proyecto dentro de la organización. Los miembros de una organización tendrán acceso a todos los proyectos dentro de la misma. También se pueden agregar miembros que tengan acceso a un determinado proyecto, sin tener acceso a toda la organización.

5. Crear un *“Cluster”* dentro de un proyecto. A modo de simplificación, pueden pensar que un “cluster” es como un servidor, que tiene cierta cantidad de memoria RAM y disco duro. Ver siguiente punto por más detalles.

MongoDB Atlas (4/4)

Instructivo (cont):

6. A la hora de crear el *cluster*, seleccionar *tier* M0, que es el *tier* gratuito, compartido (*shared*) y dispone de 512 MB de espacio en disco. Además se deberá seleccionar AWS como *Cloud Provider* y us-east como región. Lo importante es que la base de datos esté en la misma región geográfica que Vercel, así la comunicación entre ambos es más rápida. Notar que sólo se podrá crear un *cluster* gratuito por proyecto.

7. Crear un usuario/contraseña para la base de datos [Sección: *Database Access*]. No confundir con el usuario creado en el paso 1.

8. Habilitar acceso a la base de datos desde cualquier IP [Sección: *Network Access*]. La IP que debería quedar habilitada es “0.0.0.0/0”.

9. Obtener datos de acceso a la base de datos (*connection string*). Pegar esto en su archivo `.env`.