



Coding Bootcamp

Sprint 3



Temario



Temario

- ¿Qué es Deployment?
- Ngrok.
- Deployment @ Heroku.
- Deployment @ Vercel.
- MongoDB Atlas.



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 ver 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.



Ngrok



Ngrok (1/2)

Ngrok es una herramienta que genera **URLs públicas** que apuntan a una carpeta local en nuestra computadora.

Es útil para **testear**. Por ejemplo, para mostrarle una prueba a un cliente o para probar un sitio en otro dispositivo (ej: un celular).

Genera URLs con el formato: <https://59f8fa46.ngrok.io>.

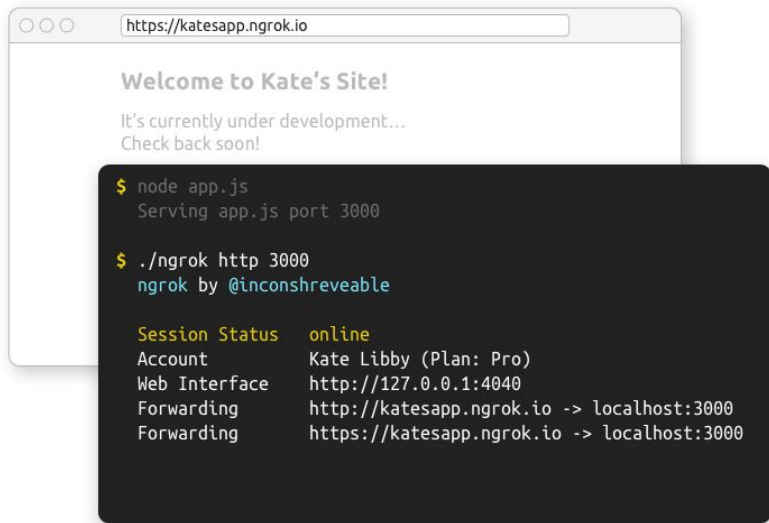


Ngrok (2/2)

Una vez que Ngrok haya sido instalado, la forma de uso es la siguiente.

Supongamos que tenemos una aplicación corriendo en el puerto 3000.

Para hacer pública dicha aplicación se corre el comando: `ngrok http 3000`



👉 Notar que la aplicación estará disponible sólo mientras la terminal esté abierta.



Deployment @ Heroku

Heroku (1/5)



Heroku 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.



DEVELOPERS

Focus on your apps

Invest in apps, not ops. Heroku handles the hard stuff — patching and upgrading, 24/7 ops and security, build systems, failovers, and more — so your developers can stay focused on building great apps.

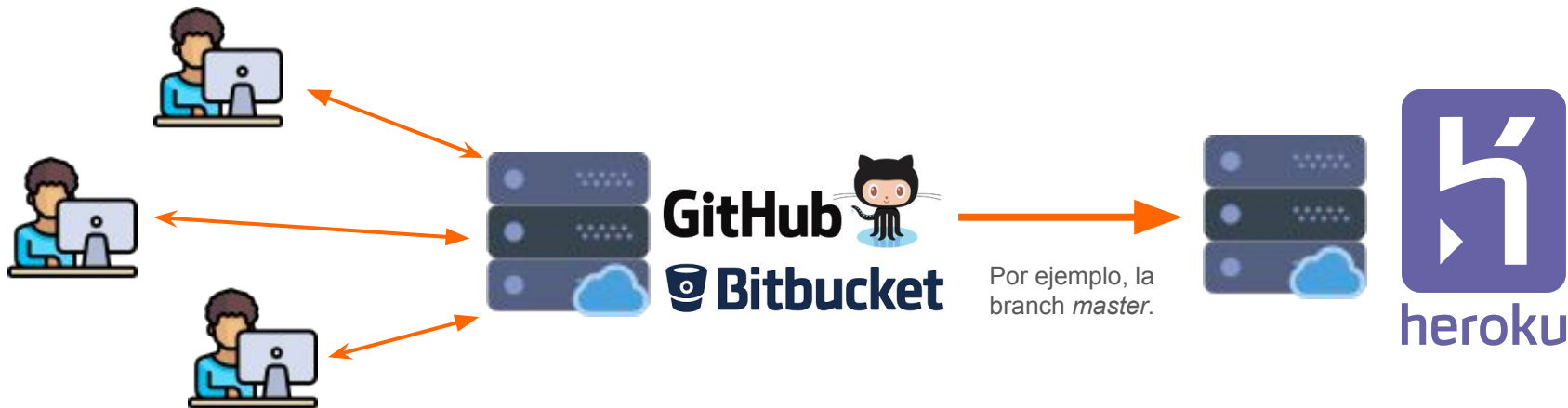
[SIGN UP FOR FREE](#)

[Explore the Heroku Platform](#)

Heroku (2/5)



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





Heroku (3/5)

Heroku ofrece un **servicio gratuito**, ideal para testear aplicaciones, aunque tiene ciertas **limitantes**:

- Sólo tiene 512 MB de RAM.
- Se “duerme” luego de 30 minutos de inactividad.
- No incluye base de datos (de forma integrada).

Sin embargo, tiene **ventajas** como:

- Facilidad de uso.
- Creación de URLs con HTTPS. Ej: <https://mi-app.herokuapp.com>.
- Configuración de dominios propios (ej: miempresa.uy) [sin costo].
- Add-ons.



Heroku (4/5)

Heroku **no permite guardar** (en sus servidores) **archivos subidos por los usuarios** de la aplicación. Por ejemplo, en el caso de un blog, las fotos que se suben al crear un artículo.

Esta característica de Heroku es lo que se conoce como [Heroku Ephemeral Filesystem](#).

Por lo tanto, si necesita que los usuarios puedan subir archivos, es indispensable utilizar un servicio externo para el guardado de los mismos. Ej: [Amazon S3](#).

Heroku (5/5)



Algunas **consideraciones** a la hora de “deployar” una aplicación **Node.js/Express** en Heroku:

- En la carpeta raíz de la aplicación, crear un archivo llamado **Procfile** (sin extensión) y colocar dentro del mismo el siguiente texto: `web: node server.js`
- Dado que no existe un archivo `.env` en el repo (en GitHub), las **variables de entorno** se deben crear desde el dashboard de Heroku, en la sección *Settings > Config Vars*.
- La **base de datos** y el contenedor de **archivos** se debe configurar de forma independiente. La forma más sencilla es vía un [Add-On](#).
- Puede llegar a ser **caro** si el proyecto crece: <https://www.heroku.com/pricing>.



Deployment @ Vercel

Vercel (1/5)



Develop. Preview. Ship.

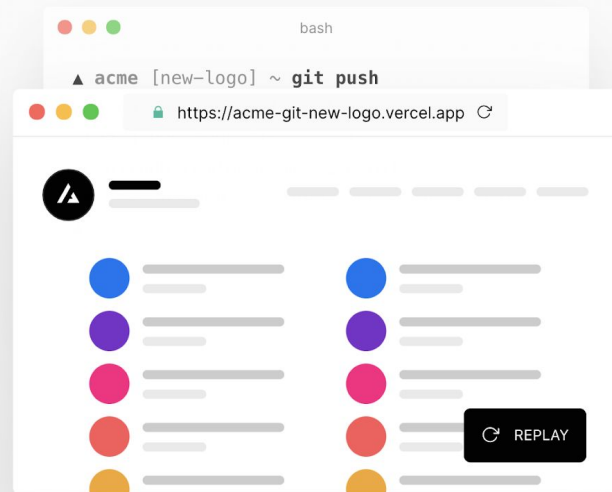
Vercel is the optimal workflow for frontend teams.

All-in-one: Static and Jamstack deployment,
Serverless Functions, and Global CDN.

Deploy Free

Contact Sales

Takes 15 seconds





Vercel (2/5)

Vercel (antes llamado Zeit) es un servicio de **hosting** con una “filosofía” similar a Heroku: el desarrollador debe enfocarse en el desarrollo de la aplicación, no en tareas de SysAdmin o DevOps. Pero tiene algunas características propias:

- Está construido “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 (3/5)

Similitudes con Heroku:

- No incluye base de datos ni cloud storage (hay que conseguirlo a parte).
- Creación de **URLs** con HTTPS. Ej: <https://mi-app.now.sh>.
- Configuración de **dominios propios** (ej: miempresa.uy) [sin costo].
- En lugar Add-ons, se tienen **Integrations**.
- Es necesario configurar las **variables de entorno** desde el *dashboard* de Vercel.

Vercel (4/5)



Al igual que con Heroku, una de las facilidades de Vercel es poder linkear 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 (5/5)

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": "@now/node-server" },
    { "src": "/public/**/*", "use": "@now/static" }
  ],
  "routes": [
    {
      "src": "/css/(.*)",
      "dest": "/public/css/$1"
    },
    {
      "src": "/(.*)",
      "dest": "/server.js"
    }
  ]
}
```

Con ****/*** se indica: “cualquier archivo en cualquier sub-carpeta”. Ver detalles sobre este formato [aquí](#) y [aquí](#).

Esto es una expresión regular.

Con este JSON le indicamos a Vercel cómo debe ejecutar los archivos que componen la aplicación. Ver [detalles](#).



MongoDB Atlas

MongoDB Atlas (1/2)



MongoDB Atlas

Move faster with a cloud MongoDB service. Built for agile teams who'd rather spend time building apps than managing databases. Available on AWS, Azure, and GCP.

Start free




Already have an account? [Log in here](#) →

Cloud Provider & Region

Choose your preferred cloud provider and the region nearest to clients














AWS, N. Virginia (us-east-1) >

Select a cloud provider to see its region availability.



Configure a **free tier cluster** by first selecting a region labeled with **FREE TIER AVAILABLE** then choose the M0 option in the Cluster Tier below.

★ recommended region ⓘ

NORTH AMERICA	EUROPE	ASIA
<div> N. Virginia (us-east-1) ★ FREE TIER AVAILABLE</div>	<div> Ireland (eu-west-1) ★</div>	<div> Tokyo (ap-northeast-1)</div>
<div> Ohio (us-west-1) ★</div>	<div> London (eu-west-2)</div>	<div> Seoul (ap-northeast-2)</div>
<div> N. California (us-west-1)</div>	<div> Frankfurt (eu-central-1) ★ FREE TIER AVAILABLE</div>	<div> Singapore (ap-southeast-1)</div>
<div> Oregon (us-west-2) ★</div>		<div> Mumbai (ap-south-1)</div>
<div> Montreal (ca-central-1)</div>	<div> São Paulo (sa-east-1)</div>	



MongoDB Atlas (2/2)

- [MongoDB Atlas](#) es un servicio de la empresa MongoDB Inc.
- Es **gratuito** para BD de hasta 512 MB.
- Está **construido “arriba” de AWS, Azure y GCP**. Atlas permite elegir en cuál de estas tres nubes alojar nuestra BD.
- Al crear una BD en Atlas, obtendremos un **string de conexión** que se deberá agregar en el método `mongoose.connect` de la aplicación.
Ejemplo: `mongodb+srv://<user>:<pass>@my-cluster-uf345.mongodb.net/<db-name>?retryWrites=true` .
- Para hacer llamadas a la BD desde Vercel o Heroku, será necesario agregar “0.0.0.0/0” como una [whitelist entry](#). Esto habilita que cualquier IP pueda llamar a Atlas.
- Docs: <https://docs.atlas.mongodb.com/getting-started>.