**UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

**FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

**Escuela Profesional de Ingeniería de Software**

**ServiceHomeStore**

**Estrategias de despliegue**

Versión 2.0

**Lima, enero del 2021**

**Historial de revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha de revisión** | **Versión** | **Descripción** | **Autor** |
| 15/01/2021 | 1.0 | Se realizó la documentación de estrategias de despliegue | * Aquino Limache Cristian. * Quinteros Peralta Rodrigo * Tirado Julca Juan Jose |
| 29/01/2021 | 2.0 | Se completo el documento de estrategias de despliegue | * Aquino Limache Cristian. * Quinteros Peralta Rodrigo * Tirado Julca Juan Jose |

**ÍNDICE**

[1.INTRODUCCIÓN](#_diu8bfjkvxo4) **4**

[2.OBJETIVOS](#_fcadekvbz552) **4**

[3.ALCANCE](#_umsz33pe6xb5) **4**

[4.RECURSOS DE HARDWARE](#_a4z0mm6jq30g) **4**

[5.ESTRATEGIAS](#_frmb4rnz5vdq) **5**

[5.1. Despliegue con Heroku](#_9tbh3wmpt2sq) 5

[¿Qué es?](#_hro0bx3c5zb7) 5

[¿Cómo aplicarlo?](#_ojqv029hmtlq) 5

[Sincronización con la rama](#_6qxzwoxvlp7e) 7

[Configurar la Base de Datos](#_ot6b6lqk8x55) 8

[Funcionamiento de la aplicación](#_m65kdasj2xma) 9

[5.2. Despliegue con Amazon Web Services(AWS)](#_gw0daoakli4y) 10

[¿Qué es?](#_tez1hmj781ds) 10

[¿Cómo aplicarlo?](#_frhe4x275cj7) 10

[Crear y conectar el repositorio](#_28f0ki5w9rb2) 10

[Confirmar la configuración de creación](#_n7rjv5ijozc7) 12

[Guardar e implementar](#_ax4c6fgmjdmk) 13

# **1.INTRODUCCIÓN**

El presente documento explicará cómo el proyecto ServiceHomeStore(SHS) pasará a la fase de despliegue por medio de dos plataformas como Heroku y Amazon (AWS). De eligieron estas plataformas debido a su popularidad y también por su plan gratuito.

# **2.OBJETIVOS**

* Brindar dos plataformas en la nube para realizar el despliegue del proyecto.
* Validar que todas las funcionalidades del sistema están disponibles para su uso.

# **3.ALCANCE**

* Presentar una estrategia de implementación y despliegue relacionada con la configuración y despliegue del sistema.
* Conseguir un mejor manejo del despliegue del sistema por medio de distintas plataformas.
* Hacer uso de dos estrategias de despliegue como Heroku y AWS.

# **4.RECURSOS DE HARDWARE**

|  |  |  |
| --- | --- | --- |
| **Dato** | **Valor mínimo** | **Valor recomendado** |
| Procesador | 2GHz | 2GHz |
| Memoria RAM | 8GB | 16GB |
| Tamaño Almacenamiento | 300GB | 700GB |

# **5.ESTRATEGIAS**

## **5.1. Despliegue con Heroku**

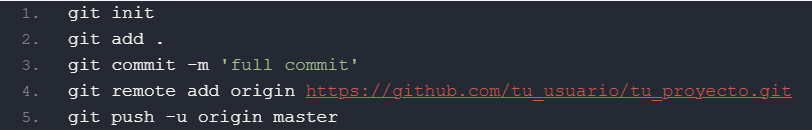
### **¿Qué es?**

Heroku es una plataforma de servicios integrados para implementar y ejecutar aplicaciones modernas que brinda como experiencia al desarrollador un enfoque centrado en la aplicación para la entrega de software, integrando las herramientas y flujos de trabajo más populares en la actualidad.

### **¿Cómo aplicarlo?**

La aplicación al estar realizada en laravel, nos apoyaremos en comandos propios de laravel y el repositorio de github.

1. Ejecutar comandos php artisan test y php artisan dusk en la consola de laravel, estos comandos se utilizan para verificar que nuestra aplicación está pasando las pruebas.
2. En github seguimos las siguientes instrucciones para subir el proyecto:



Debemos cambiar la url por la de nuestro repositorio donde se encuentre el proyecto.

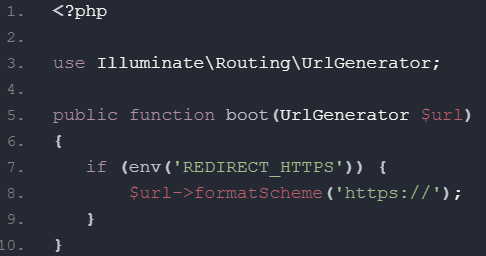
1. Luego añadimos un archivo llamado Procfile en la raíz del proyecto el cual contendrá

web: vendor/bin/heroku-php-apache2 public/

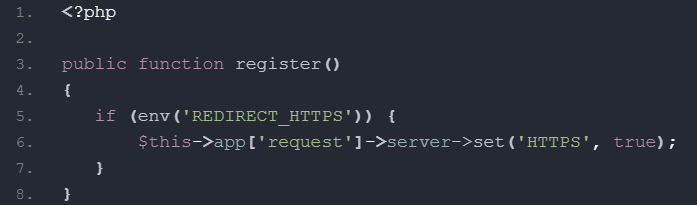
Una vez realizado, subir los cambios al repositorio github

1. Heroku nos brinda una url de tipo https por lo cual, en el proyecto debemos de asegurar que genere este tipo de ruta en todas las rutas.

Primero importamos la clase Illuminate\Routing\UrlGenerator

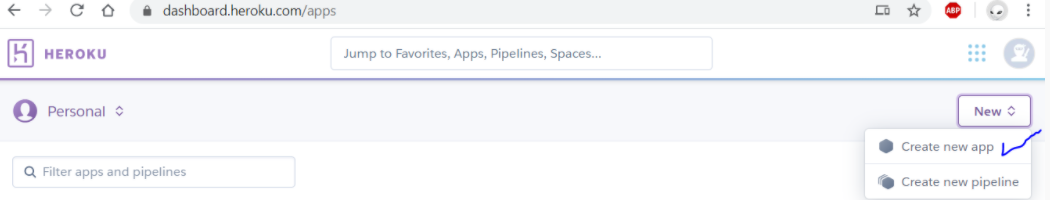
Luegos nos dirigimos al archivo app/Providers/AppServiceProvider.php y agregamos lo siguiente en el método boot:  


Y luego en el método register:

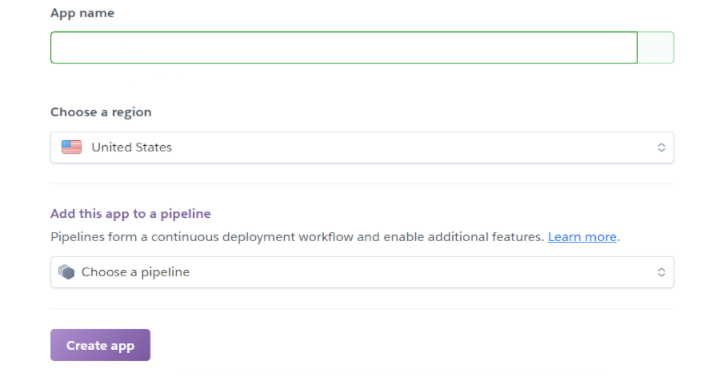


Procedemos a guardar los cambios a nuestro repositorio github.

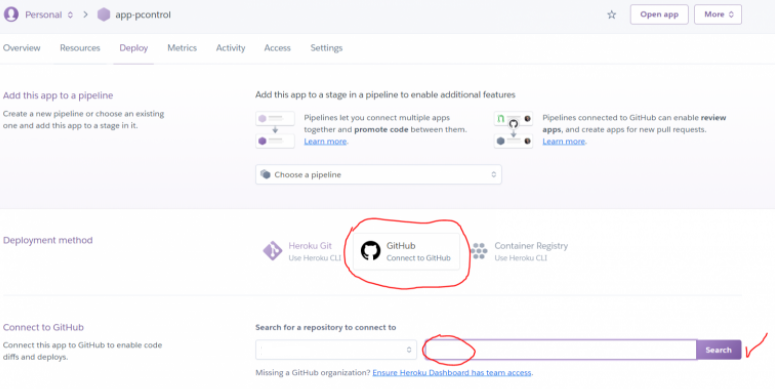
1. Ingresamos a Heroku por medio de nuestra cuenta previamente creada y hacemos click en new y luego en Create new app



Luego en la sección Create new app ingresamos el nombre de nuestra app y el resto lo dejamos por defecto, procedemos a hacer click en Create app:

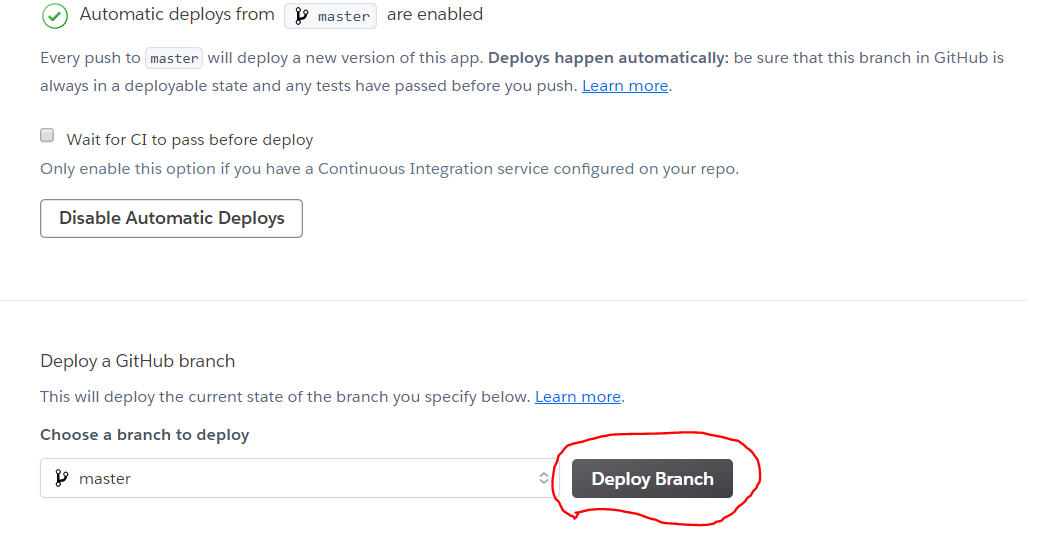


1. En la sección deploy, hacemos click en github, luego colocamos el nombre del repositorio y hacemos click en buscar. Una vez que visualizamos nuestro repositorio hacemos click en connect.



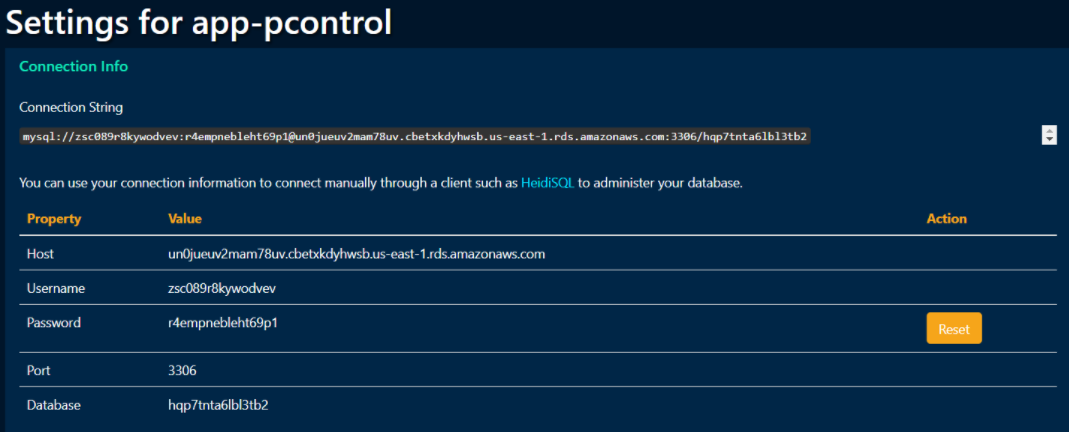
### **Sincronización con la rama**

Seleccionamos la rama que queremos sincronizar la cual será main, luego hacemos click en Enable Automatic Deploys, luego hacemos click en deploy branch.



### **Configurar la Base de Datos**

Para configurar la base de datos nos dirigimos a la opción Resources, en el buscador Maria DB o MySQL, luego hacemos click en en el botón Provision. Luego hacemos click en la base de datos que elegimos, posteriormente nos mostrará los datos necesarios para conectar la base de datos.



Por último ingresamos a la opción Config Vars y colocamos los datos que obtuvimos previamente



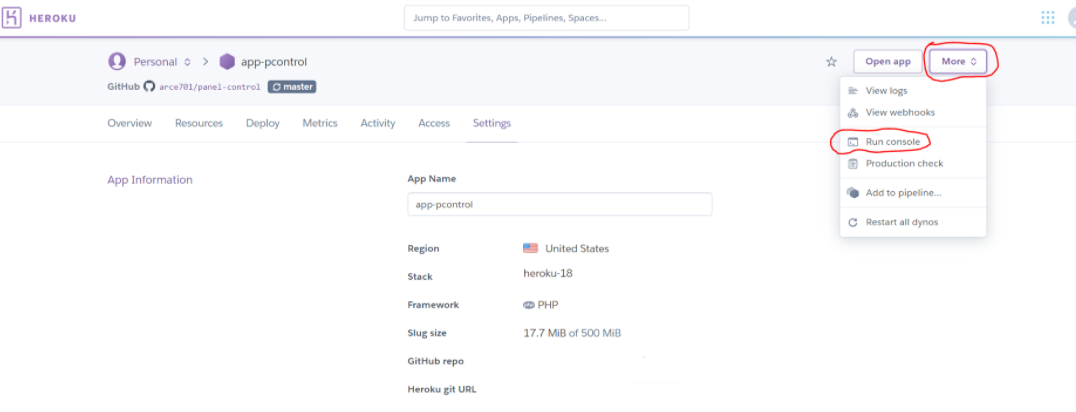
### **Funcionamiento de la aplicación**

Para que la aplicación funcione tenemos que restaurar la base de datos, la cual realizaremos por medio de migraciones ejecutando en la consola los comandos:  
composer update

composer dump-autoload

Por último en el panel de heroku ejecutamos la consola y el comando:

php artisan migrate --seed



## **5.2. Despliegue con Amazon Web Services(AWS)**

### **¿Qué es?**

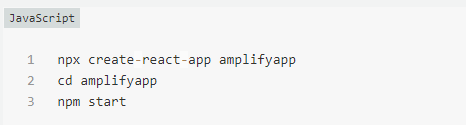
Amazon Web Services (AWS) nos permite realizar el despliegue en la nube, brinda una cantidad de servicios como la de almacenamiento a la gestión de distancias, imágenes virtuales, desarrollo de aplicaciones, etc. Los servicios que ofrece AWS pueden ser usados para autónomos, pequeñas y grandes empresas. A continuación se presentarán las herramientas que ofrece Amazon Web Services en estas categorías:

* Cloud computing: Amazon EC2 es usada dentro de los servicios de computación en la nube de Amazon.
* Base de datos: Para este caso es usado Amazon RDS, permite que permanezcan en la nube distintas bases de datos como MySQL, PosgreSL, Oracle, etc.
* Creación de redes virtuales: Con la ayuda de Amazon VPC podemos crear redes privadas virtuales a través de la nube.

### **¿Cómo aplicarlo?**

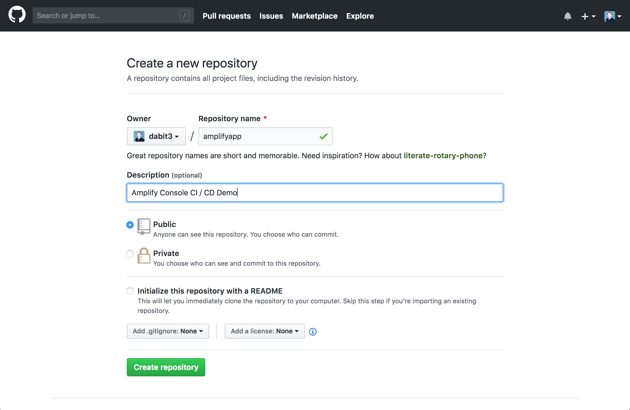
### **Crear y conectar el repositorio**

Primero debemos crear e iniciar el repositorio. Podemos hacerlo mediante el comando create-react-app. Instalar los paquetes necesarios haciendo uso del comando del sistema.



Se creará un repositorio en GitHub, en caso de que se tenga creado no es necesario, en donde enviaremos el código. Pero primero se necesitaría una cuenta de GitHub

1. Crear un nuevo repositorio en GitHub



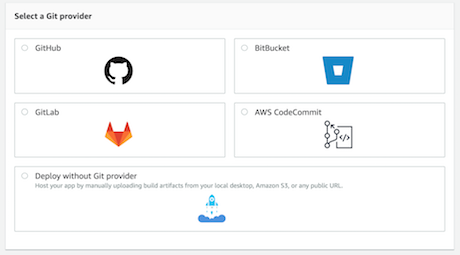
1. Luego iniciaremos Git, y enviaremos al nuevo repositorio que creamos en GitHub, ejecutando el siguiente comando.



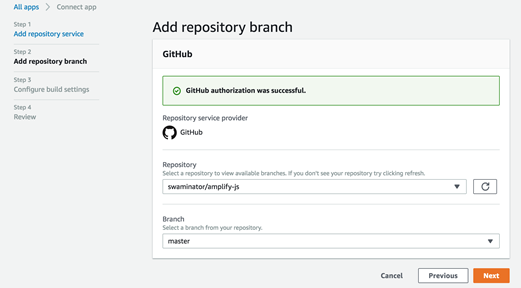
1. Para conectar el repositorio, inicie sesión en la consola de Amplify y seleccione Get started (Introducción) en Deploy (Implementar).



1. Luego conectamos los repositorios en este caso GitHub. Amplify accede al repositorio mediante las claves de implementación instaladas solo en un repositorio específico.

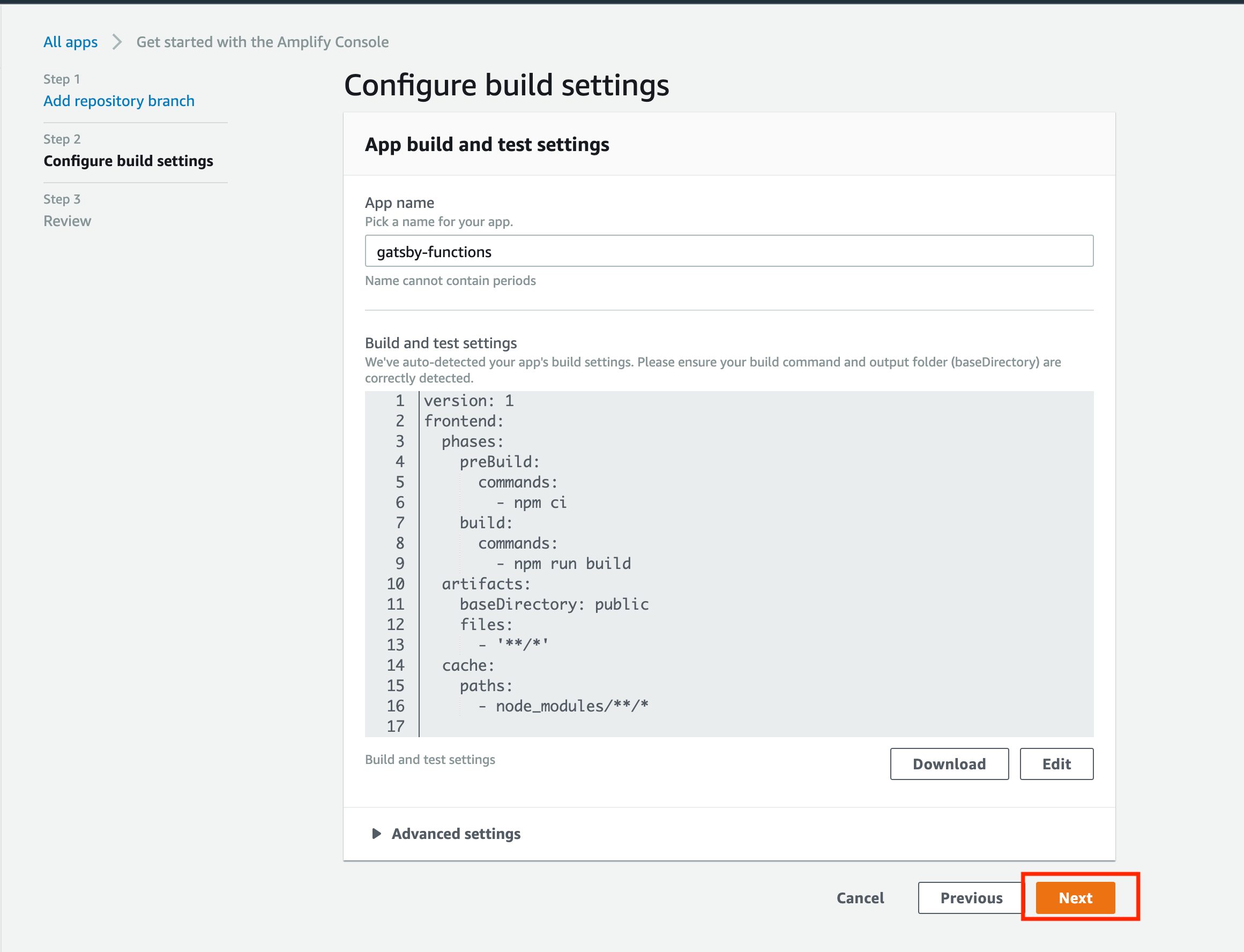


1. Elegimos un repositorio y luego seleccionamos una de las ramificaciones correspondientes para crear e implementar.



### **Confirmar la configuración de creación**

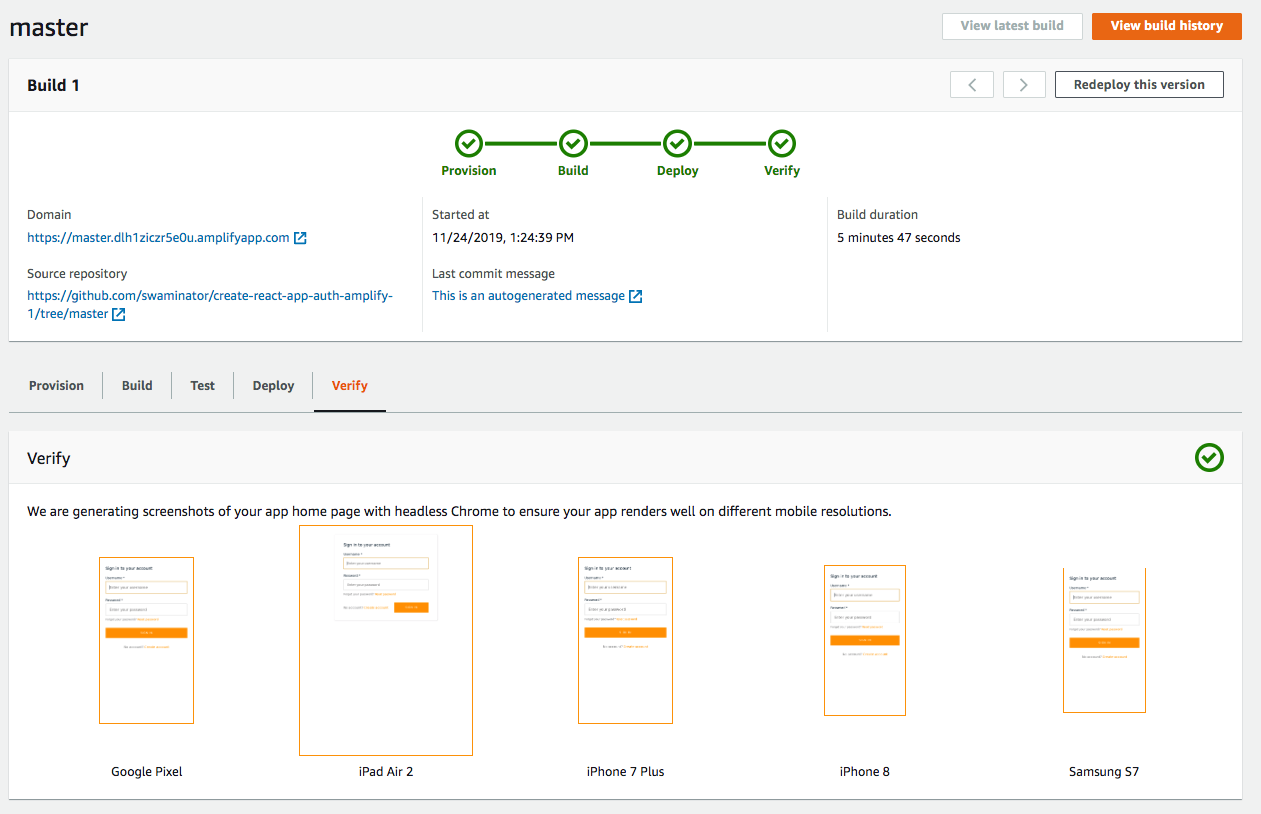
1. Para la ramificación seleccionada, Amplify inspecciona el repositorio a fin de detectar automáticamente la secuencia de comandos de creación que debe ejecutarse. Por lo cual debemos de confirmar las configuraciones de creación y luego hacemos click en siguiente.



### **Guardar e implementar**

1. Revisamos todos los ajustes necesarios para tener en cuenta que todo se haya configurado correctamente. Si es así, seleccionamos Save and deploy, para implementar la aplicación web en una red de entrega de contenido global.

El tiempo de demora se relaciona con el tamaño de la aplicación.

****