**HEROKU**

Heroku le permite implementar, ejecutar y administrar aplicaciones escritas en Ruby, Node.js, Java, Python, Clojure, Scala, Go y PHP.

Los mecanismos de dependencia varían según los lenguajes: en Ruby se usa un , en Python a , en Node.js a , en Java a y así sucesivamente.Gemfilerequirements.txtpackage.jsonpom.xml

El código fuente de su aplicación, junto con el archivo de dependencia, debe proporcionar suficiente información para que la plataforma Heroku construya su aplicación, para producir algo que se pueda ejecutar.

**Saber ejecutar**

No es necesario realizar muchos cambios en una aplicación para ejecutarla en Heroku. Un requisito es informar a la plataforma sobre qué partes de su aplicación se pueden ejecutar.

Para otras aplicaciones, es posible que deba declarar explícitamente lo que se puede ejecutar. Esto se hace en un archivo de texto que acompaña a su código fuente: un archivo Procfile. Cada línea declara un tipo de proceso, un comando con nombre que se puede ejecutar en la aplicación compilada.

**Implementación de aplicaciones**

Git es un potente sistema de control de versiones distribuido que muchos desarrolladores utilizan para administrar y versionar el código fuente. La plataforma Heroku utiliza Git como el medio principal para implementar aplicaciones (hay otras formas de transportar su código fuente a Heroku, incluso a través de una API).

Cuando crea una aplicación en Heroku, asocia un nuevo control remoto Git, normalmente denominado, con el repositorio Git local para su aplicación.heroku

Como resultado, la implementación de código es solo la familiar, pero para el control remoto:git pushheroku

**Creación de aplicaciones**

Cuando la plataforma Heroku recibe el origen de la aplicación, inicia una compilación de la aplicación de origen. El mecanismo de compilación suele ser específico del lenguaje, pero sigue el mismo patrón, normalmente recuperando las dependencias especificadas y creando los activos necesarios (ya sea tan simple como procesar hojas de estilo o tan complejo como compilar código).

**Ejecución de aplicaciones en dinamómetros**

Heroku ejecuta aplicaciones ejecutando un comando que especificó en el Procfile, en un dinamómetro que se ha precargado con su slug preparado (de hecho, con su liberación, que extiende su slug y algunos elementos aún no definidos: config vars y complementos).

Piense en un dinamómetro en ejecución como un contenedor Unix ligero, seguro y virtualizado que contiene el slug de su aplicación en su sistema de archivos.

Generalmente, si implementa una aplicación por primera vez, Heroku ejecutará 1 dinamómetro web automáticamente. En otras palabras, arrancará un dinamómetro, lo cargará con su slug y ejecutará el comando que ha asociado con el tipo de proceso web en su Procfile.

Usted tiene control sobre cuántos dinamómetros se están ejecutando en un momento dado. Dado el ejemplo anterior de Procfile, puede iniciar 5 dynos, 3 para la web y 2 para los tipos de proceso de cola, de la siguiente manera:

$heroku ps:scale web=3 queue=2

**Config vars**

La configuración de una aplicación es todo lo que es probable que varíe entre entornos (staging, producción, entornos de desarrollador, etc.). Esto incluye servicios de copia de seguridad como bases de datos, credenciales o variables de entorno que proporcionan información específica a la aplicación.

Heroku le permite ejecutar su aplicación con una configuración personalizable: la configuración se encuentra fuera del código de su aplicación y se puede cambiar independientemente de él.

La configuración de una aplicación se almacena en config vars. Por ejemplo, a continuación, se indica cómo configurar una clave de cifrado para una aplicación:

heroku config:set ENCRYPTION\_KEY=my\_secret\_launch\_codes

Adding config vars and restarting demoapp... done, v14

ENCRYPTION\_KEY: my\_secret\_launch\_codes

Debido a que Heroku administra y ejecuta aplicaciones, no es necesario administrar sistemas operativos u otra configuración interna del sistema. Los dinamómetros únicos se pueden ejecutar con su entrada/salida conectada a su terminal local. Estos también se pueden utilizar para llevar a cabo tareas de administración que modifican el estado de los recursos compartidos, por ejemplo, la configuración de la base de datos, tal vez periódicamente a través de un programador.

Conectar Heroku con MongoDB

Crea una cuenta en Heroku y MongoDB: Si aún no tienes una cuenta en Heroku, regístrate en Heroku. Luego, asegúrate de tener una base de datos MongoDB disponible. Puedes usar un servicio de MongoDB en la nube como MongoDB Atlas o configurar una instancia local de MongoDB.

Configura tu aplicación de Heroku:

Asegúrate de que tienes la Herramienta de línea de comandos de Heroku (Heroku CLI) instalada en tu sistema.

Inicia sesión en tu cuenta de Heroku utilizando el comando heroku login si aún no lo has hecho.

Crea tu aplicación en Heroku:

Utiliza el comando heroku create para crear una nueva aplicación en Heroku si aún no tienes una.

Configura las variables de entorno: Es importante mantener la información de configuración, como las URL de conexión a la base de datos, segura y fuera del código fuente. Heroku te permite configurar variables de entorno para esto. Puedes hacerlo a través de la CLI de Heroku o desde el panel de configuración de la aplicación en el sitio web de Heroku.

Para una base de datos MongoDB, deberás definir una variable de entorno que contenga la URL de conexión a la base de datos. Por ejemplo, puedes usar la URL proporcionada por MongoDB Atlas o la URL de tu base de datos local. Por ejemplo: ***heroku config:set MONGODB\_URI=your\_mongodb\_uri***

Configura tu aplicación para usar MongoDB: En tu aplicación, debes utilizar una biblioteca o controlador de MongoDB para conectarte a la base de datos. Dependiendo del lenguaje de programación que estés utilizando, puedes encontrar bibliotecas específicas. Por ejemplo, en Node.js, puedes usar la biblioteca mongoose.

Instala las dependencias necesarias, incluyendo la biblioteca de MongoDB en tu proyecto.

En tu código, utiliza la variable de entorno que configuraste en el paso 4 para obtener la URL de conexión a MongoDB y establecer la conexión.

Despliega tu aplicación en Heroku: Utiliza el comando git push heroku master (o el nombre de tu rama principal) para desplegar tu aplicación en Heroku.

Verifica y escala tu aplicación: Verifica que tu aplicación se está ejecutando correctamente en Heroku utilizando el comando heroku open. Si todo está funcionando como se espera, puedes escalar tu aplicación según sea necesario.