

Despliegue de una aplicación web REST en la nube

Objetivos

- Aprender a realizar el despliegue de una aplicación web con servicios REST en una plataforma como servicio (PaaS). En este caso se hará uso del proveedor Heroku [<https://www.heroku.com>].
- Ejercitarse en la realización de pruebas de carga.
- Aprender a instalar y a hacer uso de una herramienta de monitoreo.

Herramientas

1. Eclipse [<http://www.eclipse.org/downloads/>]
2. Heroku [<http://heroku.com/>]
3. Plugin de Heroku para eclipse [<http://marketplace.eclipse.org/content/heroku-eclipse-integration>]
4. New Relic [<http://newrelic.com/>]

Pre-Requisitos

1. Para poder desarrollar este laboratorio es indispensable que tenga instalado eclipse [<https://www.eclipse.org/downloads/>].
2. A continuación debe crear una cuenta en Heroku [<https://id.heroku.com/signup>], para esto basta con registrar su correo.
3. Luego es necesario instalar el plugin de heroku para eclipse. Este procedimiento se explica claramente en el siguiente link [<http://eclipse-plugin.herokuapp.com/>]. Hasta el paso "Adding your SSH Key to Heroku" (incluyendo la realización de este último).

Nota: Si el laboratorio lo está desarrollando en la máquina virtual que se le asignó para el curso no es necesario instalar el plugin, ya se encuentra instalado en dichas máquinas.

Descripción de la Aplicación

La aplicación que se usará para el desarrollo de este laboratorio está orientada a la inscripción de participantes a un concurso. La aplicación actual ofrece dos servicios principalmente:

1. Inscripción de un participante al concurso.
2. Obtener todos los participantes del concurso.

La empresa dueña del concurso desea dejar de usar su propia infraestructura para realizar el despliegue de este tipo de aplicaciones, ya que estas aplicaciones están consumiendo demasiados recursos, los cuales son necesarios para las aplicaciones core de la empresa.

Es por esto que se desea realizar el despliegue de la aplicación de concurso en una *plataforma cloud*, como *Heroku*.

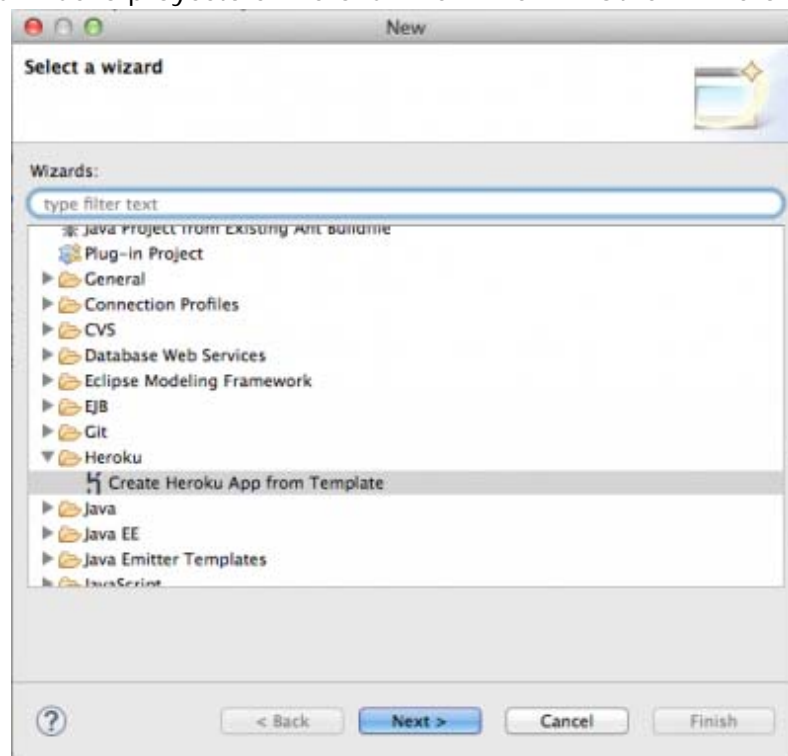
Descripción de los Servicios REST

A continuación encontrará un pdf con la especificación de los servicios REST.

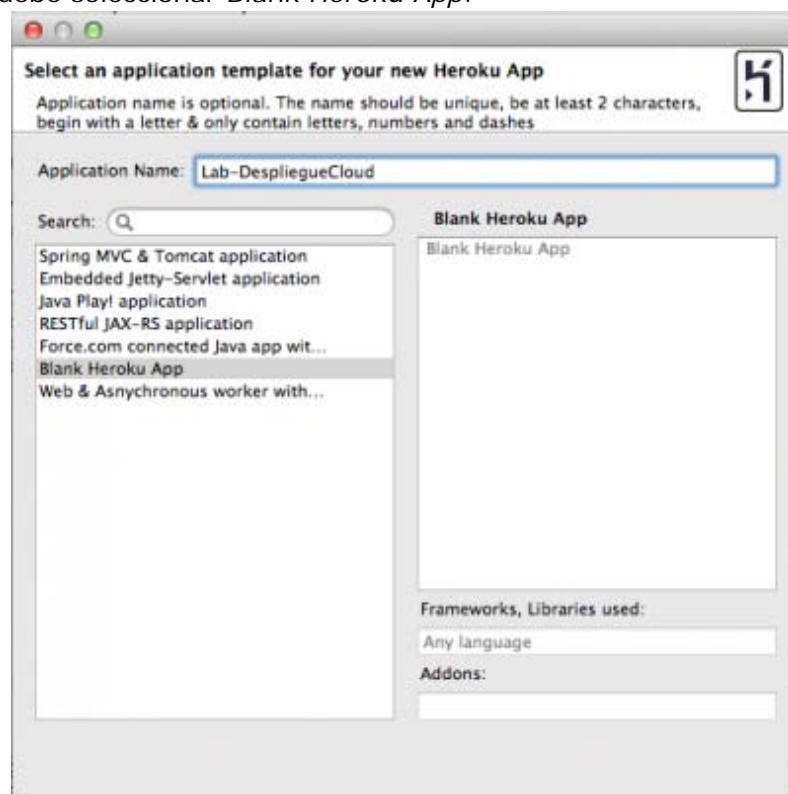
Esta especificación muestra: las urls de cada servicio, los parámetros y la respuesta. [Descargar PDF](#)

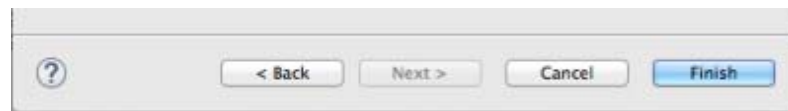
Despliegue de la Aplicación

1. Debe descargar el código base del siguiente: Repositorio [<https://github.com/maveme/Lab-DespliegueCloud.git>]
2. Abra Eclipse y cree un nuevo proyecto en heroku. *File → New → Other → Heroku*.

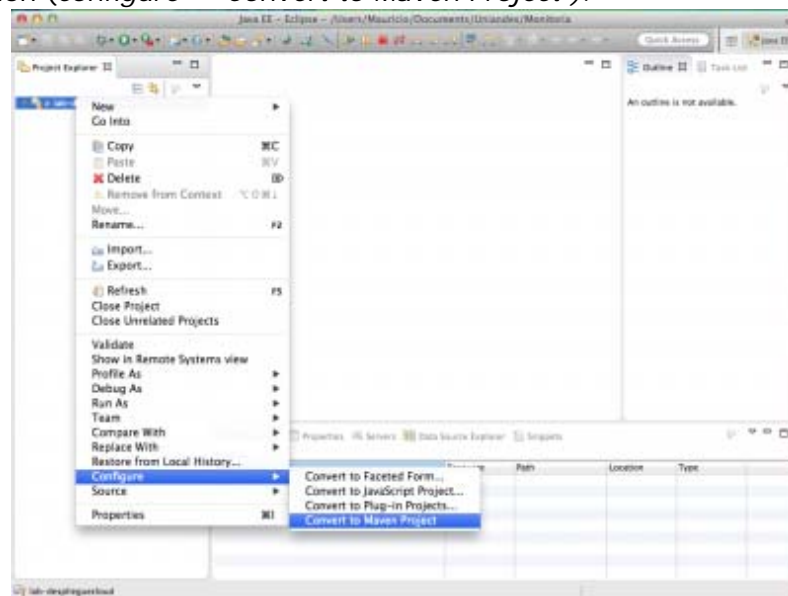


3. Ahora debe colocarle un nombre a la aplicación (el nombre debe ser único, por ejemplo: lab-despliegue-<login-uniandes>) y posteriormente seleccionar el tipo de proyecto que desea crear. En este caso debe seleccionar *Blank Heroku App*.

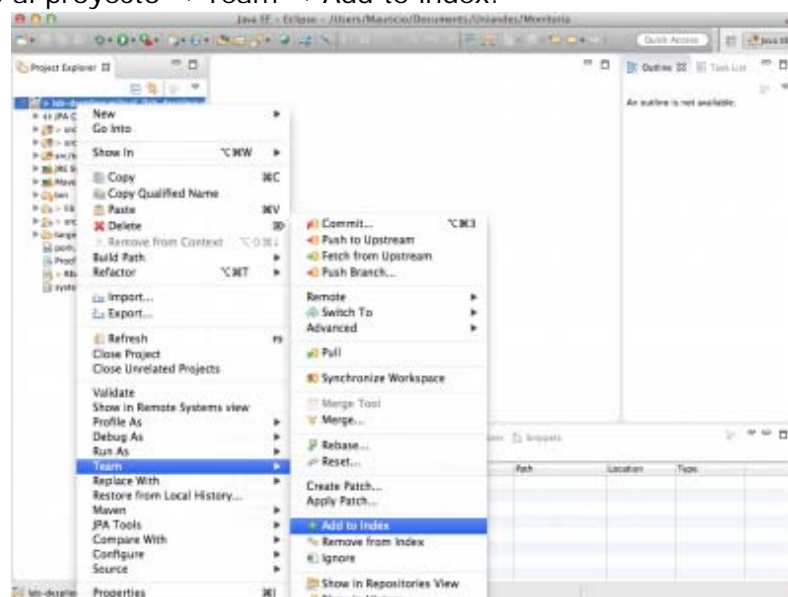




4. Ahora debe copiar todo el código y las carpetas que descargó en el paso 1, y copiarlos en el directorio que haya seleccionado para crear la aplicación de Heroku.
5. Debe refrescar la vista de la aplicación en heroku.
6. Luego debe configurar el proyecto para que sea un proyecto maven, para esto, debe darle click derecho a la aplicación (*configure* → *Convert to Maven Project*).

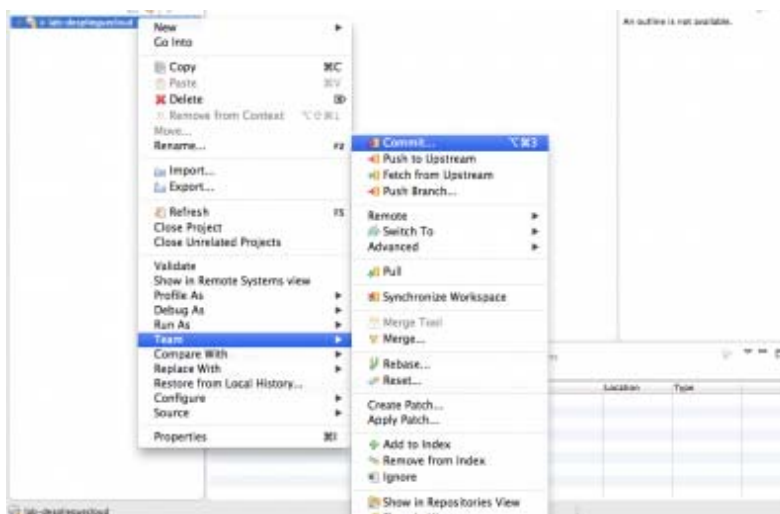


7. Luego de esto, vamos a subir los cambios que hemos realizado de manera local para que estos pasen al ambiente de producción que tenemos desplegado en heroku. Primero debemos agregar los archivos a nuestro repositorio Git (ya viene por defecto en las aplicaciones de heroku). Para esto le damos click derecho al proyecto → Team → Add to index.

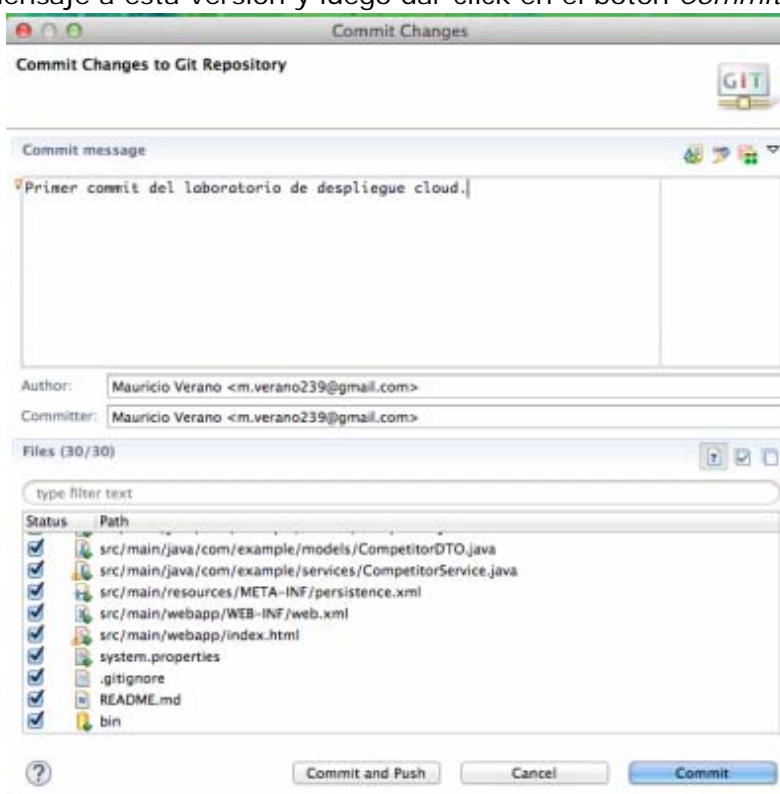


8. Ahora es necesario hacer commit de los cambios, para esto debe darle click derecho al proyecto → Team → Commit



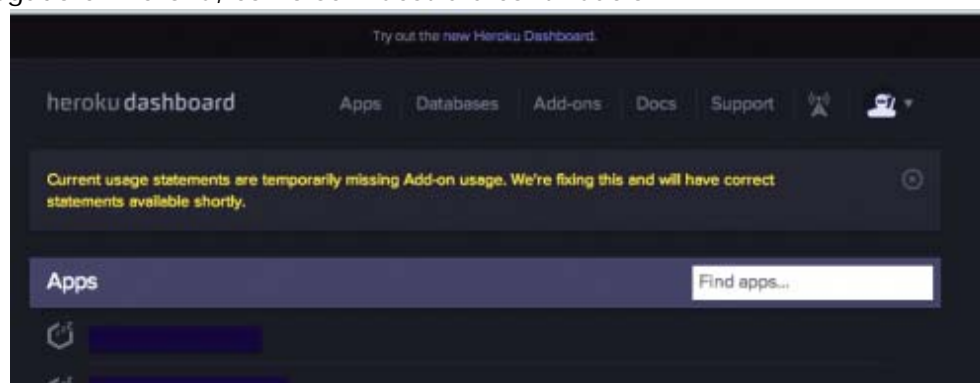


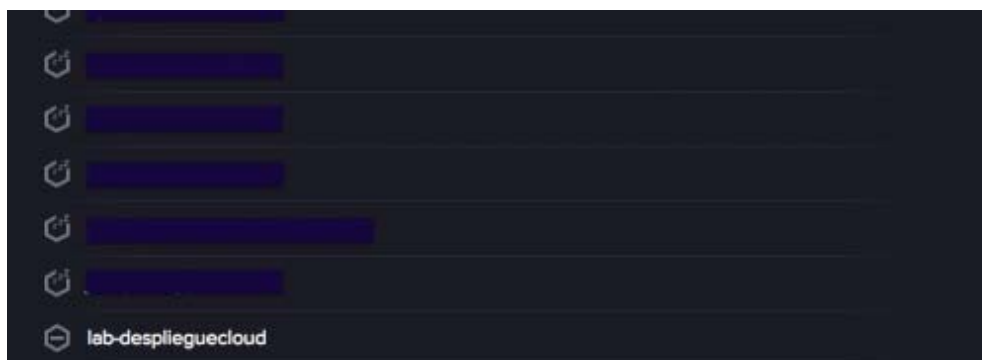
9. Deben agregar un mensaje a esta versión y luego dar click en el botón *Commit and Push*



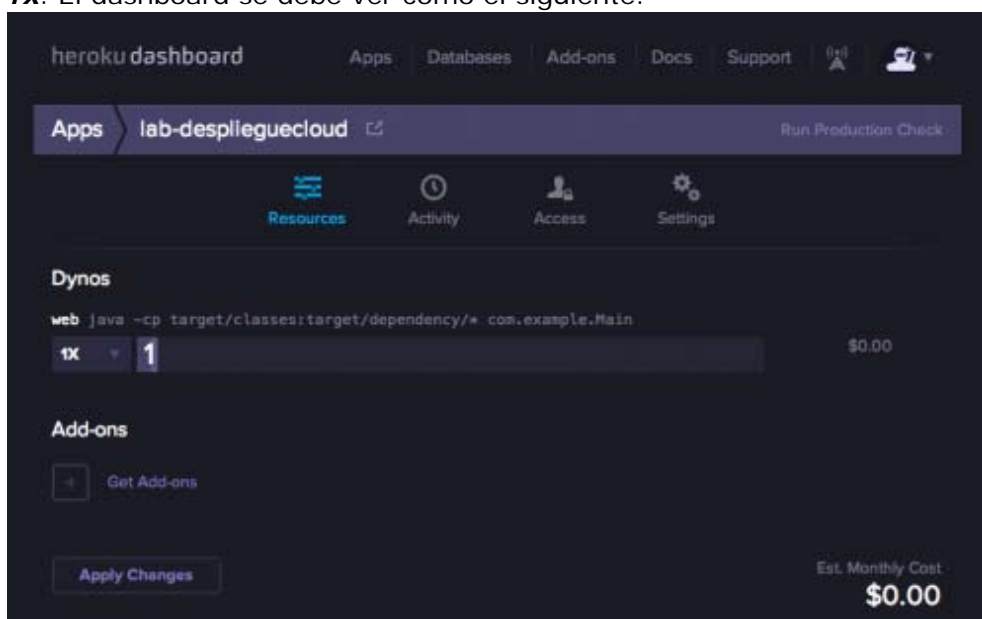
10. Ahora deben acceder a la página de Heroku [https://dashboard.heroku.com/apps]

11. En esta página deberán hacer login y posteriormente les aparecerá una lista con todas las apps que hayan desplegado en heroku, como se muestra a continuación:





12. Es necesario darle click en el nombre de nuestra aplicación, en este caso "*lab-desplieguecloud*". En el dashboard que les aparece deben verificar que la cantidad de Dynos esté en **1**, y que el tipo de Dyno sea de **1x**. El dashboard se debe ver como el siguiente:

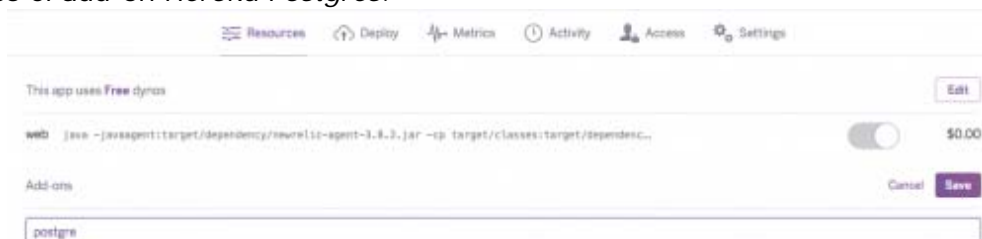


Hasta el momento hemos realizado el despliegue de la aplicación en Heroku, pero aún nos falta la configuración de la base de datos y de la herramienta de monitoreo que usaremos.

Configuración de la Base de Datos

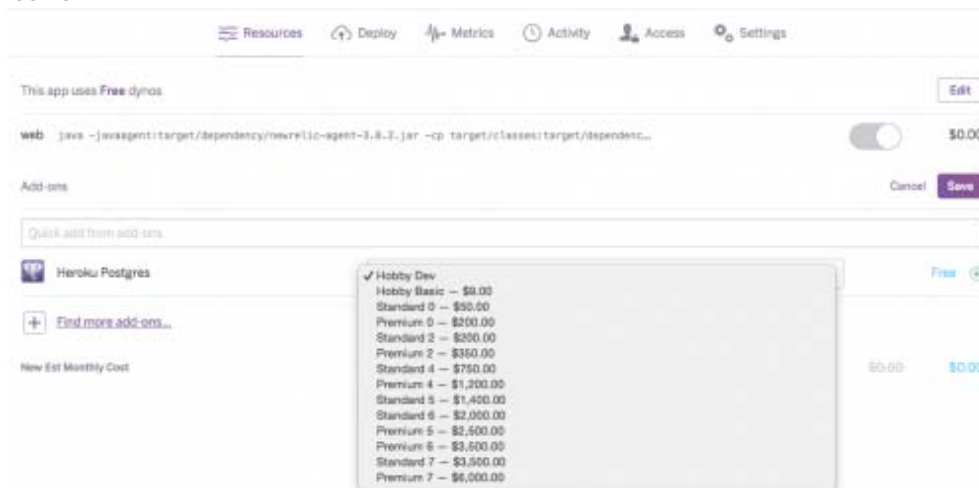
La plataforma que estamos usando, cuenta con plugins para instalar distintos tipos de bases de datos, tanto relacionales como no relacionales. Heroku trabaja por defecto con bases de datos relacionales Postgres, por lo tanto vamos a proceder a configurarla.

1. Debemos acceder al dashboard de nuestra aplicación y vamos a darle click al botón **+** que está al lado izquierdo del link *find more add-ons...*. Aquí se mostrará una barra de búsqueda con los distintos add-ons que ofrece Heroku. En este caso vamos a buscar por la palabra *postgres* y seleccionamos el add-on *Heroku Postgres*.

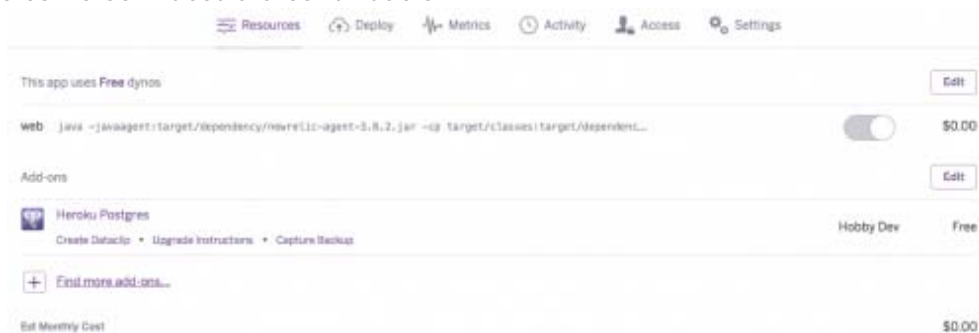




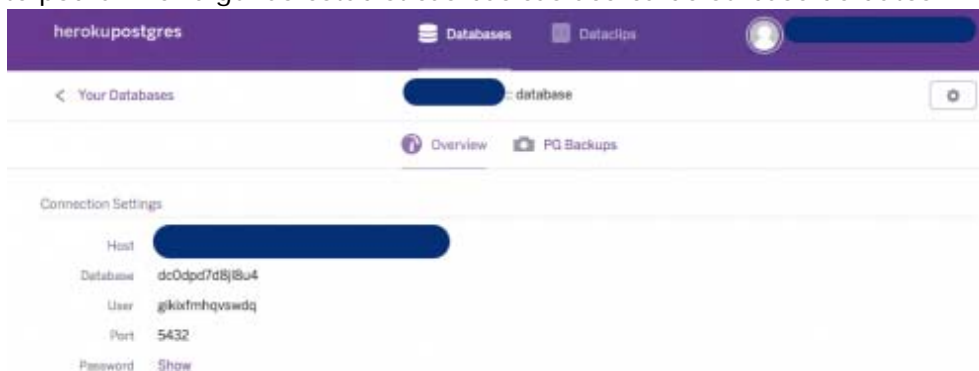
- Ahora debemos seleccionar el tipo de base de datos que queremos agregar a nuestra aplicación, en este caso vamos a usar una base de datos gratuita de desarrollo *Hobby Dev*. Luego deberán darle click al botón *save*.

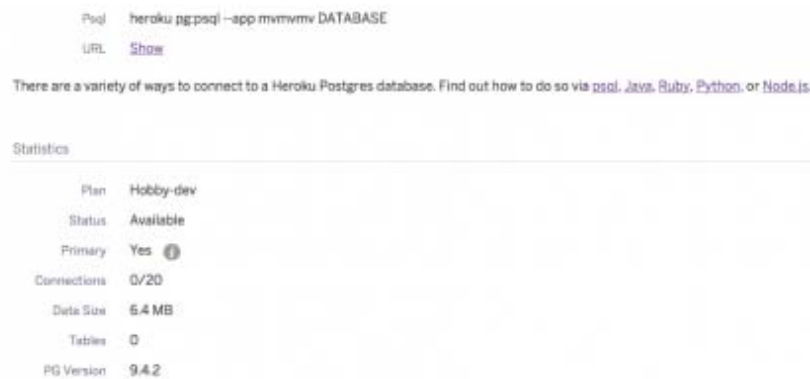


- Luego dentro del dashboard de su aplicación en Heroku deberían ver que se agregó el add-on de la base de datos como se muestra a continuación:



- Ahora deben darle click al add-on de Postgres y les aparecerá la lista de bases de datos Postgres que tienen creadas en Heroku.
- En este punto deben revisar que en listado de bases de datos esté una con el mismo nombre de la aplicación de Heroku, si no es el caso, deberá darle click al botón *create database*.
- Deberán darle click a la base de datos que acabaron de crear.
- A continuación les aparecerán las credenciales para poderse conectar a su base de datos y adicionalmente podrán ver algunas estadísticas básicas acerca de su base de datos.





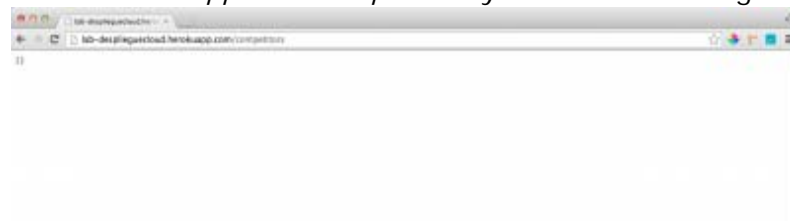
8. Teniendo las credenciales a la mano, deben ir a su aplicación en eclipse y modificar la unidad de persistencia (persistence.xml) y agregar los datos de SU base de datos.
9. Deben agregar la url: esta se construye a partir del: *Host + puerto + database*. Esta información la obtienen de las credenciales de la base de datos. Les debe quedar algo de la siguiente forma:

```
jdbc:postgresql://<host>:<port>/database?ssl=true&sslfactory=org.postgresql.ssl.NonValidatingFactory"
```

10. Posteriormente deben agregar el nombre del usuario y la contraseña de la base de datos.
11. El persistence.xml les debe quedar similar a este pero con sus credenciales:

```
<?xml version='1.0' encoding='UTF-8'?>
<persistence.xml xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence http://xmlns.jcp.org/xml/ns/persistence/persistence_2_0.xsd" >
  <persistence-unit name="DatabaseMysql" transaction-type="LOCAL" >
    <jdbc-driver-class>org.postgresql.Driver</jdbc-driver-class>
    <jdbc-url>jdbc:postgresql://<host>:<port>/<database></jdbc-url>
    <jdbc-user><user></jdbc-user>
    <jdbc-password><password></jdbc-password>
  </persistence-unit>
  <persistence-unit name="DatabaseMysql" transaction-type="JTA" >
    <jta-data-source>java:comp/env/jtaDataSource</jta-data-source>
  </persistence-unit>
</persistence.xml>
```

12. Finalmente deben hacer commit de los cambios realizados. (Al igual que en el numeral 8 de la sección anterior).
13. Luego de que hayan subido los cambios, desde un browser acceden a la url de su aplicación. *http: <Nombre Aplicación>.herokuapp.com/competitors* y deberían ver lo siguiente:

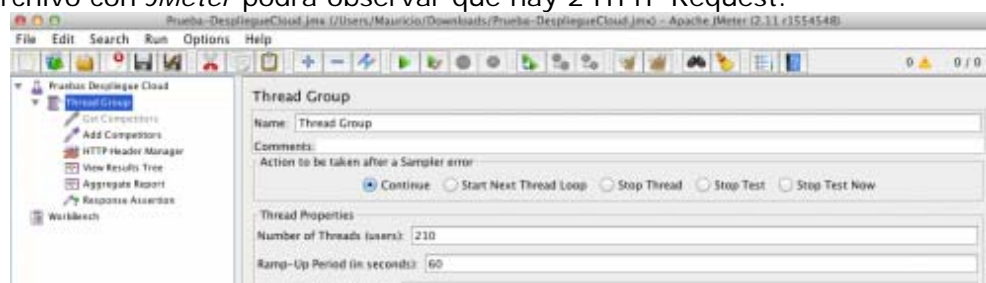


Ejecución de pruebas JMeter

Dirjase al directorio de instalación de Apache JMeter (preguntarle al monitor). Dentro de dicho directorio de instalación, ubique la carpeta bin y ejecute el archivo ApacheJMeter.jar.

A continuación se deberá ejecutar una prueba de carga usando *JMeter*. El archivo de prueba puede ser descargado desde aquí.

Al abrir dicho archivo con *JMeter* podrá observar que hay 2 HTTP Request:





Uno para el servicio de crear competidores y otro para el servicio de dar todos los participantes. Uno de los HTTP Request está de color gris (deshabilitado), por lo tanto no se está ejecutando. Se hará la prueba de carga sólo usando el request que está habilitado.

Nota: Se sugiere habilitar el HTTP Request deshabilitado una vez haya culminado las actividades que siguen. Para esto, basta con darle click derecho al HTTP Request y luego seleccionar la opción *enable*.

Actividades

Ejecute 6 escenarios de prueba, en cada ejecución:

1. Modifique la cantidad de usuarios del HTTP request habilitado como sigue:

- 300
- 600
- 3000
- 6000
- 30000
- 60000

1. Haga clic en el elemento "Summary Report (Reporte Resumen)" para consultar los resultados.
2. Guarde todas las métricas del reporte (tiempo mínimo, tiempo máximo, tiempo promedio, % error y throughput) en un archivo excel.
3. Luego de ejecutar todas las pruebas, realice un análisis de los resultados obtenidos.

Entregables

Deben subir a Sicua+ archivo .zip con lo siguiente:

1. Documento excel con los resultados obtenidos luego de ejecutar las pruebas.
2. Análisis de los resultados obtenidos en las pruebas.

Opcional

- Configuración Herramienta de Monitoreo
- Configuración Herramienta de Monitoreo (sin tarjeta de crédito) [https://sistemas.uniandes.edu.co/~isis2503/dokuwiki/doku.php?id=laboratorios:desplieguerest-cloud&#new_relic_sin_tarjeta_de_credito]

Problemas frecuentes al usar el plugin de eclipse

En algunos casos se presentan problemas al realizar la conexión ssh, la siguiente es un posible

procedimiento para solucionar dicho inconveniente:

1. Verifiqué la clase ssh que ha registrado en el plugin de eclipse, si está vacía deberá crear una y guardarla. Si el problema persiste siga los siguientes pasos:
1. Descargar un Cliente VPN, pueden usar <http://swupdate.openvpn.org/privatetunnel/client/privatetunnel-win-2.3.exe> o <https://www.surfeasy.com/> [<http://swupdate.openvpn.org/privatetunnel/client/privatetunnel-win-2.3.exe>] o [<https://www.surfeasy.com/>]
2. Si ya les aparece en heroku que crearon la app, deben ir a heroku.com y darle click al nombre de su app. Luego deben darle click a la pestaña de "settings" y ahí copiar la url que les aparece en *Git URL*.
3. En Eclipse seleccionan file→import→Git→Projects from Git y le dan click al botón next
4. Luego deben seleccionar Clone URI y le dan click al botón next
5. En el campo de URI, pegan la URL que copiaron de git y luego le dan click al botón next
6. Es posible que les pregunte que si desean agregar el dominio de heroku a los "known host" le dan click al botón si.

laboratorios/laboratorio1.txt · Última modificación: 2015/07/02 12:57 por rg.meneses81