

Ambientes no propietarios

Heroku - Cloud Computing

Edwin Salvador

14 de julio de 2015

Sesión 15

1 Cloud Computing

- Tipos
- Modelos de servicio
 - SaaS
 - PaaS
 - IaaS

2 Heroku

Cloud Computing

- Un paradigma que permite ofrecer servicios de computación a través de Internet.
- Empezó en proveedores de servicio de Internet a gran escala, como Google, Amazon AWS, Microsoft.
- Tenían una infraestructura tan potente que no se utilizaba todo el tiempo que decidieron alquilar el hardware a manera de maquinas virtuales.
- La tendencia es la computación como servicio, que la computación se vuelva un servicio básico y se cobre solo por el uso de los recursos.
- Beneficios:
 - Agilidad (rápida implementación)
 - Costo (Solo paga por lo que usa)
 - Escalabilidad y elasticidad (recursos por demanda)
 - Rendimiento (balance de carga)
 - Seguridad (altos niveles de seguridad)
 - Mantenimiento (prácticamente nulo por parte del usuario)

- Desventajas:
 - Alta de dependencia de proveedores.
 - Datos sensibles en servidores remotos
 - Servicios muy especializados pueden tardar años en estar disponibles.

1 Cloud Computing

- Tipos
- Modelos de servicio
 - SaaS
 - PaaS
 - IaaS

2 Heroku

- Ventajas:

- Computación como servicio
- Gestionada y mantenida por terceras personas.
- El cliente se puede enfocar en el negocio.
- Costos efectivos.
- Tamaño preciso.
- El acceso a servicio a través de Internet.

- Desventajas:

- Seguridad
- Datos y procesos de varios clientes se mezclan.
- Los usuarios no saben que otros procesos están corriendo.
- Pérdida de control

- Ventajas:
 - Facilidad de seguridad
 - Infraestructura como servicio gestionada para un solo cliente
 - Mayor control
- Desventajas:
 - Caducidad de hardware
 - Costos
 - Personal especializado

- Ventajas:
 - Usan nube privada pero cambian a pública cuando es necesario.
 - El cliente es propietario de una parte pero otras partes son compartidas aunque de manera controlada.
- Desventajas:
 - No es simple la compartición de datos entre la nube pública y privada.
- Presenta las mismas ventajas y desventajas que las nubes privadas y públicas.

Nube Comunitaria

Cloud Computing

- Este modelo se define como aquel que se organiza con la finalidad de cumplir una función común (seguridad, política).
- Son administradas por las organizaciones que la constituyen o terceras partes.

1 Cloud Computing

- Tipos
- Modelos de servicio
 - SaaS
 - PaaS
 - IaaS

2 Heroku

SaaS (Software as a Service)

- Software como servicio
- Está en la capa más alta, es una aplicación completa ofrecida como servicio
- Una sola instancia del software corre en el servidor y atiende a varios usuarios.
- Las aplicaciones son utilizadas por el navegador web.
- Evita la instalación de software local.
- Control muy limitado.
- Gmail, Sharepoint, Salesforce.com CRM, Gaikai, Microsoft Office 365.

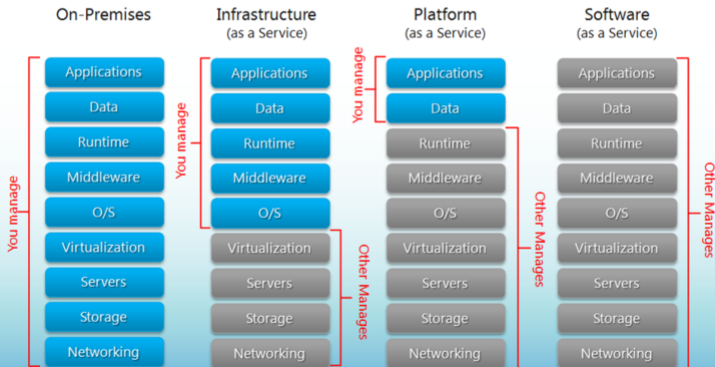
PaaS (Platform as a Service)

- Plataforma como servicio
- Es la capa del medio
- Abstracción de ambiente de desarrollo (SO, lenguaje de programación, BD, etc)
- El usuario controla las aplicaciones pero no la infraestructura.
- Google App Engine, Microsoft Azure, **Heroku**, Amazon Elastic MapReduce, Aneka.

IaaS (Infrastructure as a Service)

- Infraestructura como servicio (hardware as a Service)
- La capa inferior
- Utiliza la virtualización para brindar sistemas de almacenamiento, red, procesador, RAM, etc.
- Balance de carga de trabajo, aumento de capacidad durante horas pico (elasticidad)
- Amazon Web Services (El más popular), Rackspace Cloud, NeCTAR Research Cloud, Joyent Cloud, GoGrid, FlexiScale.

Separation of Responsibilities



1 Cloud Computing

- Tipos
- Modelos de servicio
 - SaaS
 - PaaS
 - IaaS

2 Heroku

- Es una plataforma como servicio de computación en la Nube que soporta distintos lenguajes de programación.
- Propiedad de Salesforce.com.
- La base del sistema operativo es Debian Ubuntu.
- Características: (**Dynos = contenedores Linux**)
 - **Elasticidad y crecimiento.** La capacidad asignada a una aplicación se puede cambiar en cualquier momento.
 - **Tamaño.** diferentes tipos de **dynos**, cada uno con diferentes capacidades de procesamiento y memoria.
 - **Routing.** Internamente los routers realizan un seguimiento de la ubicación de los Dynos que estén corriendo, y redirigen el tráfico de acuerdo a la misma.
 - **Seguimiento.** Manejador de Dynos monitorea continuamente los dynos que se estén ejecutando. Si uno falla es eliminado y creado nuevamente.
 - **Distribución y redundancia.** Los Dynos se encuentran aislados uno de otro. Los fallos en la infraestructura interna de alguno de ellos no afecta a los otros, y consecuentemente tampoco la aplicación.

Heroku en práctica

- Heroku se integra fácilmente con Github. Primero deben crear un repositorio para la aplicación que desean subir en Github y configurarlo como ya sabemos.
- Heroku maneja las aplicaciones PHP a través de **Composer**: es una herramienta de gestión de dependencias de PHP. Permite declarar las librerías de terceros que nuestro proyecto necesita y las instala/actualiza por nosotros de una manera ágil. Más info en <https://getcomposer.org/doc/00-intro.md>
- Una vez que tengamos nuestro repo, agregamos un archivo llamado `composer.json` y en este archivo escribimos `{}`. Los subimos a Github.
- Si desean desarrollar apps con otro lenguaje deben seguir los pasos respectivos aquí:
<https://devcenter.heroku.com/articles/buildpacks>

Heroku en práctica

- Ingresar a <http://www.heroku.com> y crear cuenta.
- Iniciar sesión. Clic en el (+) de la esquina superior derecha para agregar una app.
- Asignarle un nombre a su APP.
- Ir al Dashboard. Clic en Deploy. Clic en GitHub. Clic en Connect to GitHub.
- Buscar el nombre del repo para la aplicación que crearon. Clic en Connect.
- Creamos un archivo `index.php` en nuestro repo y los subimos.
- Vamos al Dashboard de Heroku y vamos al final y damos clic en Deploy Branch. Si todo está correctamente configurado podemos abrir nuestra aplicación y veremos los resultados.

Configurar MySQL en Heroku

- Heroku ha elegido a Postgres como su BDD por defecto, sin embargo brinda extensiones para utilizar otras BDD como MySQL.
- Para esto debemos seguir los siguientes pasos:
 - En la pestaña Resources del Dashboard de Heroku.
 - En la sección Add-ons clic Edit
 - Buscar ClearDB MySQL. Seleccionar la opción ignite y guardar los cambios.
 - Necesitamos **MySQL Workbench** para tener una interfaz gráfica para nuestra BDD ya que en Heroku no contamos con phpmyadmin.
 - Descargar MySQL Workbench
<http://dev.mysql.com/downloads/file.php?id=457780>

Configurar MySQL Workbench

- Agregar una nueva conexión. Asignarle un nombre.
- Hostname:
 - Ir a la pestaña Settings, Clic en Reveal config Vars
 - Copiar el valor de la variable: CLEARDB_DATABASE_URL
 - Ej: mysql://bcedf1a3f16a26:ba8dde50@us-cdbr-iron-east-02.cleardb.net/heroku_19c2ff835f2409f?reconnect=true
 - Solo copiar la parte en rojo. Este es el hostname.
- Usuario y contraseña:
 - En Resources clic en ClearDB MySQL
 - Clic en el nombre de la BDD y luego en Endpoint Information.
 - Copiar el usuario y contraseña.
 - En Schema poner el nombre de la BDD.
 - Clic en Test Connection y luego en OK
 - Abrir la BDD y tenemos la interfaz para su administración.

Conectarse a ClearDB MySQL con PHP

- Lo único diferente que necesitamos para conectarnos a nuestra BDD es lo siguiente:

```
$url = parse_url(getenv("CLEARDB_DATABASE_URL"));

$server = $url["host"];
$username = $url["user"];
$password = $url["pass"];
$db = substr($url["path"], 1);

$conn = new mysqli($server, $username, $password, $db);
```

- Y todo lo demás es igual a lo que ya conocemos con esto estamos listos para desarrollar y desplegar nuestras aplicaciones PHP en Heroku.