

Arquitectura y Plataformas de Servidores

Manejo de infraestructura a gran escala

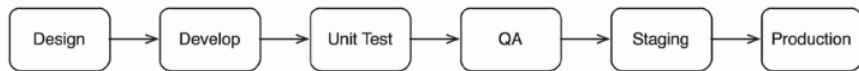
Ing. David Guevara A, Mg

UTA

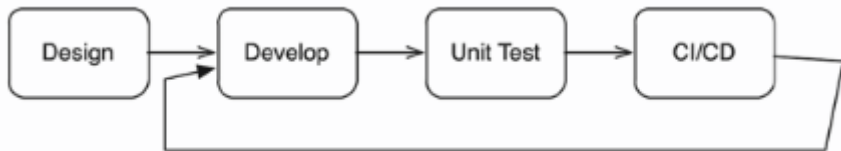
Despliegue de aplicaciones

Desde el diseño a la producción, los desarrolladores crean una infraestructura escalable, para la codificación, pruebas de test, despliegue de aplicaciones de forma rápida.

El modelo tradicional para el producción se basa en: Diseño, Desarrollo, Unidad de Test, garantía de calidad, montaje y producción.



Este modelo no es muy eficiente por lo que se propone un nuevo modelo basado en la continuidad de la integración e implementación.



Automatización del desarrollo de software

El proceso en la automatización en el desarrollo de software requiere de muchos procesos y componentes.

- Desarrollo Automatizado

- Gradle (<http://www.gradle.org>)
- Apache Ant (<http://ant.apache.org>)
- Apache Maven (<http://maven.apache.org>)
- Gnu Make (<http://www.gnu.org/software/make/>)

- Gestión en la configuración de software

- Git (<http://git-scm.com>)
- SVN (<https://subversion.apache.org>)
- Mercurial (<https://www.mercurial-scm.org/>)

- Integración continua: Confirmación de código con frecuencia, autocomprobada y compilación rápida.

- Implementación continua

- Gnu AutoConf (<https://www.gnu.org/software/autoconf/>)
- Go (<http://www.go.cd>)
- Chef (<https://www.getchef.com/chef/>)
- SaltStack (<http://www.saltstack.com>)

Automatización del desarrollo de software

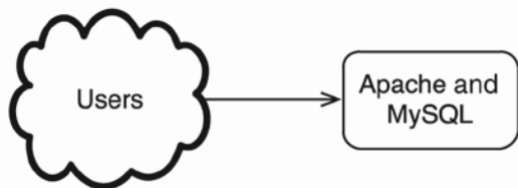
El proceso en la automatización en el desarrollo de software requiere de muchos procesos y componentes. (continuación ..)

- Gestión del cambio: Juega un papel muy importante en la gestión eficaz de la infraestructura. Es un proceso de revisión, aprobación y ejecución de cambios relacionados con los componentes de la infraestructura. se clasifican en:
 - Bajo riesgo: No afecta a la actividad de la infraestructura
 - Medio riesgo: Puede afectar a la producción. Ej Agregar nuevos VM para equilibrar la carga
 - Alto riesgo: Puede afectar a la infraestructura, Ej, Actualizar una aplicación completa, que afecte al cluster.
- Gestión de release: Es el proceso de planificación, diseño y construcción del marco necesario para llevar un producto o software en producción. Elementos a tomar en cuenta
 - Peticiones de características
 - Herramientas de packing
 - Metodología de entrega
 - Problemas y riesgos de release

- Metodología en cascada
- Metodologías ágiles
- Scrum

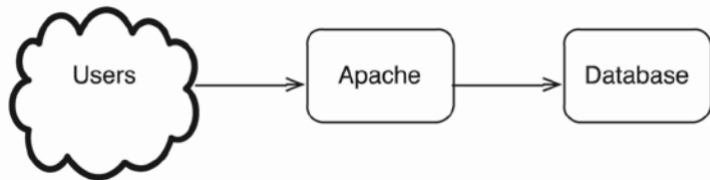
Arquitectura de un solo nivel.

- Aloja un servidor web y una base de datos
- No hay capacidades de conmutación por error
- No se recomienda a menos que no sea de misión crítica



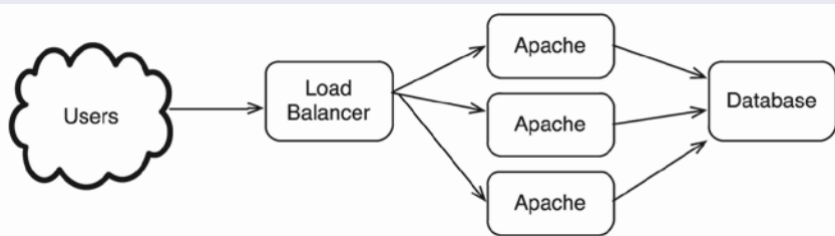
Arquitectura de dos niveles

- Agrega mejor tolerancia a fallos
- Mismos defectos que la arquitectura de un solo nivel
- No es escalable



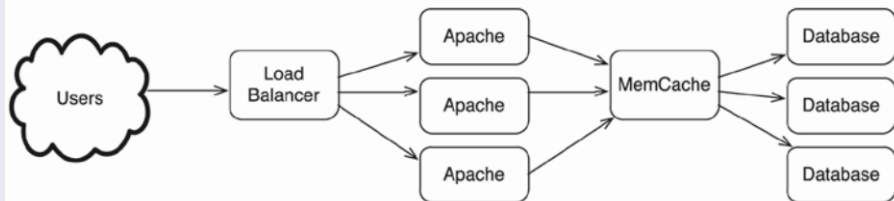
Arquitectura de tres niveles

- Introduce balanceo de carga
- Es un diseño más escalable, se puede aumentar más servidores al modelo
- Se puede configurar autoescalado
- Se puede sobrecargar la base de datos



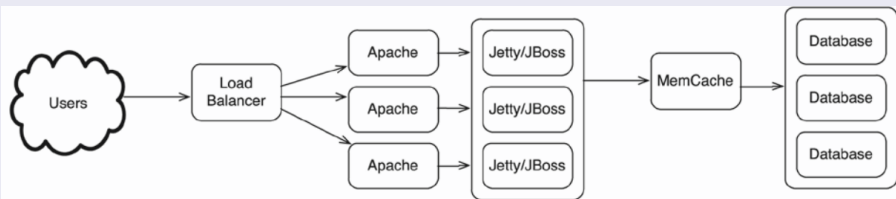
Arquitectura de cuatro niveles

- Se puede escalar el servidor web y también el servidor de base de datos (<http://memcached.org>, <http://cassandra.apache.org>)
- Es más tolerante a fallos



Arquitectura de cinco niveles

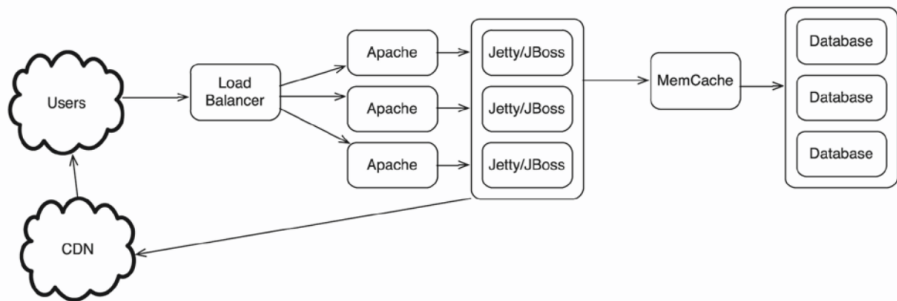
- Se agrega al servidor web y de base de datos un servidor de aplicaciones Java con Jetty o JBoss
- Se usa generalmente para distribuir contenido estático y dinámico



Arquitectura Web

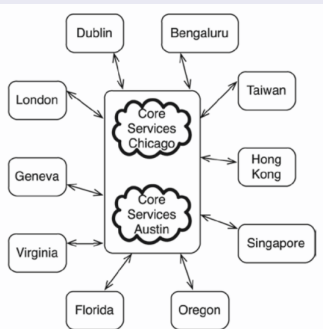
Arquitectura de seis niveles

- Se agrega una de red de entrega de contenido (content delivery network) CDN, para la descarga de contenido.
- Es útil para sitios web que publican videos o cualquier tipo de datos que requieran mucha descarga



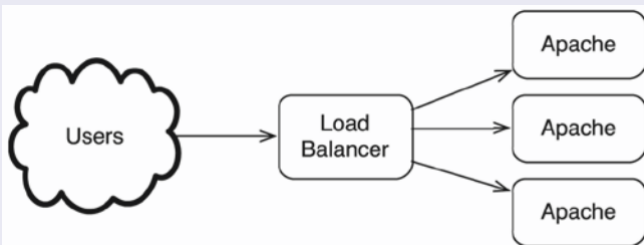
Arquitectura Global

- Permite acceso desde cualquier lugar, a través del uso de centros de datos.
- Para mejorar la eficiencia se puede crear puntos de presencia de almacenamiento en caché para dar servicios de baja latencia a los usuarios cercanos.
(<http://trafficserver.apache.org>)
- Es caro de implementar, por lo que se puede contratar un CDN de Amazon, Rackspace.



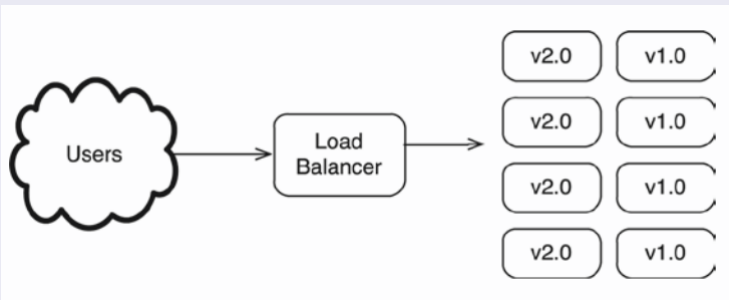
Autoescalado

- Permite aumentar o disminuir los recursos de la infraestructura para satisfacer la demanda. Ej. Netflix
- Amazon web Services o Google Cloud ofrecen autoescalado automático.



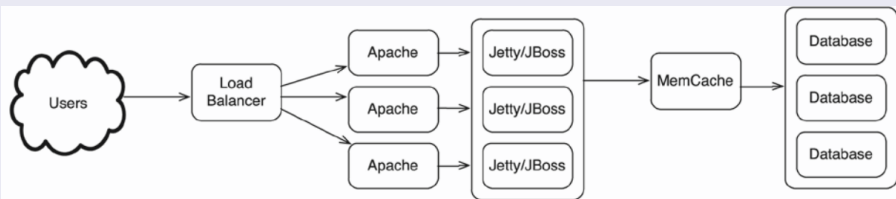
Implementaciones continuas

- Permite actualizar una pequeña porción de la infraestructura, sin afectarla toda.
- Se lo puede hacer implementando código a nuevos servidores o VM dentro de la arquitectura hasta reemplazarlos por anteriores



Arquitectura de cinco niveles

- Se agrega al servidor web y de base de datos un servidor de aplicaciones Java con Jetty o JBoss
- Se usa generalmente para distribuir contenido estático y dinámico



Ing. David Guevara A, Mg
dguevara@uta.edu.ec

Presentación desarrollada en LyX con el módulo beamer