



**Universidad**  
Zaragoza

# Proyecto Fin de Carrera

Ingeniería Informática

## Migración de infraestructura grid a tecnología cloud basada en OpenStack

**Autor:**

**Carlos Gimeno Yáñez**

**Director:**

**Rubén Vallés Pérez**

*Instituto Universitario de Biocomputación y Física de Sistemas Complejos*

**Ponente:**

**Enrique Torres Moreno**

*Arquitectura y Tecnología de Computadores*

*Departamento de Informática e Ingeniería de Sistemas*

Escuela de Ingeniería y Arquitectura Universidad de Zaragoza

2016

## Migración de infraestructura grid a tecnología cloud basada en OpenStack

### RESUMEN

El trabajo realizado en este Proyecto de Fin de Carrera, se enmarca dentro de mi trabajo en el Instituto Universitario de Biocomputación y Física de Sistemas complejos – de ahora en adelante **BIFI** - y que consta de dos partes bien diferenciadas y que se complementan para alcanzar una solución.

Por una parte, el BIFI, poseía una infraestructura de computación funcionando bajo el paradigma de “*grid computing*”. Debido a las complejas necesidades que van surgiendo en el día a día, se decidió dar el salto a una infraestructura de computación distribuida en la nube basada en OpenStack.

La primera tarea consistió en el análisis de la antigua infraestructura y el diseño de la nueva, teniendo en cuenta además las restricciones que tenía la infraestructura de computación distribuida basada en grid.

La segunda tarea consistió en el despliegue y configuración de la infraestructura de “*cloud computing*”. Para ello, se han utilizado diferentes metodologías de instalación y configuración de nodos a través de la red, y se han desarrollado un conjunto de scripts utilizando diferentes lenguajes y herramientas que permiten la automatización completa de la instalación y configuración de los nodos.

Se realizó un primer despliegue en las máquinas que el BIFI administraba en Huesca. Este primer despliegue iba orientado a dar servicio a los propios usuarios del Instituto. Sirvió además como toma de contacto con la infraestructura de computación distribuida en la nube. Sin embargo, este primer despliegue no fue perfecto, y se empezaron a ver ciertas oportunidades de mejora a tener en cuenta más adelante.

Con una demanda cada vez mayor de recursos de computación en la nube, se realizó un segundo despliegue, esta vez en las máquinas que el BIFI administraba en Zaragoza, teniendo en cuenta todas las oportunidades de mejora detectadas en el primer despliegue.

El resultado obtenido por este PFC fueron dos infraestructuras distribuidas de computación en la nube basadas en OpenStack que hasta el día de hoy, están funcionando a plena capacidad.

Como conclusiones finales, migrar de grid a la nube permitió por una parte conseguir más proyectos y usuarios al BIFI, dado que el número de aplicaciones que soporta una infraestructura de este tipo, es mucho mayor. Por otra, simplifico de manera notable la tarea de administración de la infraestructura, que era el principal objetivo de esta migración.

# Agradecimientos

---

A mis padres: Por su esfuerzo para darme la mejor educación posible y haber hecho posible el que yo haya llegado hasta este punto.

Al BIFI y a toda la gente que trabaja en el: Alfonso, Elisa, Gonzalo, Alfredo, Guillermo, Patricia, David, Alberto... Por la oportunidad de trabajar y colaborar en proyectos punteros, así como el apoyo recibido por todos y cada uno de ellos.

A Rubén: Por haberme dado la oportunidad de trabajar en el BIFI y no haber dudado de mí ni un solo momento. No cabe duda que yo no estaría redactando esto si no fuera por él.

A Enrique Torres: Por su intensa colaboración y dedicación en la redacción de este PFC.

A Alba: Por su apoyo durante todos estos años.



# Índice general

---

## Tabla de contenido

<b>1 - Introducción</b>	<b>2</b>
1.1 Alcance del documento	2
1.2 Contexto de desarrollo	2
1.3 Motivación del proyecto	2
1.4 Objetivos y alcance	3
1.5 Trabajo previo, antecedentes y posibles alternativas	3
1.6 Trabajo realizado por el autor	5
1.7 Contenido de la documentación	5
<b>2 - ¿Qué es OpenStack?</b>	<b>7</b>
<b>3 - Herramientas utilizadas</b>	<b>9</b>
<b>4 - Primer despliegue (EPSH)</b>	<b>11</b>
4.1 Objetivos	11
4.1.1 Restricciones de la infraestructura preexistente	11
4.1.2 Captura de requisitos mínimos	11
4.1.3 Casos de uso	13
4.2 Diseño y planificación temporal	13
4.2.1 Nodos de servicio	14
4.2.2 Nodos de cómputo	15
4.2.3 Almacenamiento	15
4.3 Implementación	15
4.4 Diagrama de la infraestructura	17
4.5 Pruebas y validación	18
4.6 Problemas detectados	18
4.7 Oportunidades de mejora	19
4.8 Conclusiones	19
<b>5 - Segundo despliegue (BIFI)</b>	<b>21</b>
5.1 Objetivos	21
5.1.1 Captura de requisitos mínimos	21
5.1.2 Casos de uso	22
5.2 Diseño y planificación temporal	22

# Índice general

---

5.2.1 Nodos de servicio .....	23
5.2.2 Nodos de cómputo .....	24
5.2.3 Almacenamiento .....	24
5.3 Implementación .....	24
5.4 Diagrama de la infraestructura.....	25
5.5 Pruebas y validación .....	25
5.6 Problemas detectados.....	25
5.7 Oportunidades de mejora .....	26
5.8 Conclusiones.....	26
<b>6 - Conclusiones.....</b>	<b>27</b>
6.1 Cumplimiento de objetivos .....	27
6.2 Ampliaciones y mejoras para el futuro (CESAR) .....	27
6.3 Seguimiento de la infraestructura .....	28
6.4 Incidencias .....	29
6.9 Valoraciones .....	29

# Índice de figuras

---

<i>Figura 1 - Arquitectura básica de OpenStack (<a href="http://www.solinea.com">http://www.solinea.com</a>) .....</i>	<i>7</i>
<i>Figura 2 - Planificación temporal (EPSH).....</i>	<i>14</i>
<i>Figura 3 - Esquema de la infraestructura (EPSH).....</i>	<i>17</i>
<i>Figura 4 - Planificación temporal (BIFI) .....</i>	<i>22</i>
<i>Figura 5 - Esquema de la infraestructura (BIFI).....</i>	<i>25</i>
<i>Figura 6 - Datos de uso de la infraestructura (EGI Accounting Portal).....</i>	<i>29</i>

# Índice de cuadros

---

<i>Cuadro 1 - Cuadro de requisitos de la infraestructura (EPSH) .....</i>	<i>12</i>
<i>Cuadro 2 - Cuadro de requisitos de la infraestructura (BIFI) .....</i>	<i>22</i>



# Memoria

# 1 - Introducción

---

## 1.1 Alcance del documento

El presente documento, describe el trabajo realizado por el alumno Carlos Gimeno Yáñez como PFC, titulado “*Migración de infraestructura grid a tecnología cloud basada en OpenStack*”

Este PFC consiste en la migración de una infraestructura de computación distribuida de alto rendimiento – **HTC** - basada en tecnología grid, a una infraestructura de computación distribuida en la nube, basada en **OpenStack**.

## 1.2 Contexto de desarrollo

El **BIFI** - *Instituto Universitario de Biocomputación y Física de Sistemas Complejos* -, es un instituto de investigación de la Universidad de Zaragoza, que promueve la interdisciplinariedad para afrontar los nuevos retos científicos y tecnológicos del presente. Está formado por investigadores de la Universidad de Zaragoza y de otras instituciones españolas y extranjeras, cuyo objetivo es desarrollar la investigación competitiva en las áreas de la computación aplicadas a la física de sistemas complejos y modelos biológicos.

Este PFC se enmarca en dicho instituto de investigación, en el área de Grid & Cloud Computing, área en la cual me encuentro trabajando actualmente desde hace 3 años, así como en los diversos proyectos en los que participamos en dicha área.

## 1.3 Motivación del proyecto

Hasta el momento, el BIFI ha dispuesto de una infraestructura distribuida de computación de alto rendimiento basada en tecnología grid, que le ha permitido dar servicio a un gran número de usuarios de diferentes ámbitos. Conforme el número de éstos ha ido creciendo tanto en cantidad, como en variedad de software, lo han hecho también sus necesidades, haciendo que cada vez fuera más y más complejo administrar la infraestructura.

Con el número de usuarios en aumento, aumentaba también la complejidad a la hora de administrar y mantener la infraestructura. Dado que el tiempo que pueden dedicar los administradores a administrar y mantener la infraestructura no es un factor escalable, se decidió buscar una alternativa para solucionar este problema.

Otro de los motivos para migrar esta infraestructura es que era muy rígida, ya que el número de aplicaciones que se pueden ejecutar es más limitado, tanto por el tipo de aplicaciones que se pueden ejecutar, como por limitaciones técnicas, como es el sistema operativo común de la infraestructura.

Así pues y para atajar ambos problemas, se decidió que el tipo de infraestructura que mejor solucionaba ambos problemas era una infraestructura de computación distribuida basada en la nube, ya que por una parte simplifica la tarea de administración de la misma, y por otra, gracias a la virtualización era capaz de ejecutar cualquier tipo de aplicación con la infraestructura hardware ya disponible.

# 1 - Introducción

---

## 1.4 Objetivos y alcance

El objetivo principal del PFC, es la migración completa de la infraestructura de computación distribuida basada en grid, a una infraestructura de computación distribuida en nube privada basada en OpenStack con todos sus servicios disponibles, tanto de computación, como de almacenamiento.

Una vez terminada la migración, el resultado de dicha migración será una plataforma de cloud privada, en la cual los usuarios serán capaces de crear máquinas virtuales bajo demanda para sus necesidades de cómputo, así como de nuevos dispositivos de bloques para sus necesidades de almacenamiento, o en última instancia, utilizar el almacenamiento de objetos. Todo esto podrán hacerlo mediante una sencilla interfaz web.

Asimismo, una vez terminada la migración, la tarea de administración y mantenimiento por parte de los administradores se simplificará en gran parte, ya que cada usuario será responsable de sus máquinas y de lo que instala en ellas. Los administradores solo tendrán que encargarse de monitorizar la infraestructura y comprobar que todos los servicios están funcionando correctamente.

Se han desplegado con éxito dos infraestructuras distribuidas de computación en nube privada basadas en OpenStack. Se realizó un primer despliegue que sirvió como campo de pruebas, para el segundo despliegue que estaba destinado a ser el definitivo. A día de hoy, ambos despliegues se encuentran totalmente operativos dando soporte a cientos de usuarios, tanto investigadores del propio instituto, como usuarios de varios proyectos europeos, entre ellos **EGI-Engage, Fortissimo, CloudSME o CloudfLOW**.

## 1.5 Trabajo previo, antecedentes y posibles alternativas.

En primer lugar, surge la pregunta obvia ¿Por qué migrar a una infraestructura de computación en la nube?

La respuesta es sencilla: Una infraestructura de computación distribuida basada en grid, consiste – de modo muy básico – en un nodo central al cual los usuarios mandan sus trabajos a ejecutar. Este nodo, a su vez, manda los trabajos a ejecutar a uno de los nodos de cómputo cuando haya alguno disponible.

El problema de esta aproximación es que conforme aumenta el número de usuarios, aumenta intrínsecamente sus necesidades. Por ejemplo, un usuario puede requerir una versión X de una biblioteca para lanzar su programa, mientras que otro puede requerir una versión Y, haciendo más complicada la labor de los administradores al tener que lidiar con todas estas situaciones simultáneamente.

En una plataforma de computación distribuida en la nube, la aproximación cambia radicalmente: Al usuario se le ofrecen una serie de recursos tanto computacionales – máquinas virtuales principalmente – como de almacenamiento a nivel de bloque y a nivel de objeto. A esto comúnmente se le denomina, **Infraestructura como Servicio (IaaS)**.

# 1 - Introducción

---

La ventaja (y desventaja) de esta aproximación es que los usuarios tienen total libertad para hacer lo que deseen con sus máquinas, por otra parte, si es cierto que los usuarios deben tener unas ciertas capacidades técnicas para administrar sus propias máquinas.

Con esta aproximación se consiguen dos cosas: Por una parte, los usuarios con más experiencia técnica pueden aprovechar mejor las virtudes de la infraestructura. Por otra parte, los usuarios con menor experiencia técnica pueden utilizar la infraestructura de un modo sencillo, utilizando para ello las “plantillas” que ya existen y que te permiten desplegar ciertos servicios con un par de “*clicks*”.

De cara a los administradores de la infraestructura, la administración de la misma se simplifica enormemente, ya que solo se tienen que preocupar de mantener en un estado óptimo la infraestructura con las herramientas de las que disponen, y no entrar a lidiar con problemas relacionados con el uso por parte de los usuarios.

Por último, entrando en razones puramente técnicas, es mucho más sencillo escalar una infraestructura de este tipo una vez que esté totalmente desplegada, ya que si queremos añadir más recursos de computación, únicamente tenemos que añadir más nodos de cómputo, sin necesidad de que sean del mismo fabricante.

Dicho esto al comienzo de este proyecto, se evaluaron diferentes frameworks de computación en la nube para determinar cuál era el mejor para las necesidades presentes y futuras. Los candidatos evaluados a la hora de desarrollar este PFC fueron los siguientes:

- **OpenStack:** Creado inicialmente en julio de 2010 por Rackspace y la NASA. El proyecto OpenStack pretende ayudar a las organizaciones a ofrecer servicios de computación en la nube en cualquier tipo de hardware. La primera versión salió 4 meses después, con el nombre en clave de “Austin” y con planes de liberar al público actualizaciones y nuevas funcionalidades cada pocos meses. Técnicamente, era el más avanzado hasta el momento, al contar incluso con almacenamiento de objetos, que era un requisito indispensable para el BIFI.
- **OpenNebula:** Si bien OpenNebula es más antiguo, ya que su primera versión apareció en el año 2008, a la hora de realizar este proyecto disponía de menos características que OpenStack, como por ejemplo, el almacenamiento de objetos. Asimismo, su comunidad de usuarios era menor y la implementación de nuevas funcionalidades era mucho menos frecuente que en OpenStack. Antes de desarrollar esta memoria, estaba por detrás en funcionalidades respecto a OpenStack, ya que carecía entre otras cosas del almacenamiento de objetos.

Se desplegó una pequeña infraestructura de pruebas de ambos sistemas con el objetivo de evaluar ambos, y al final se decidió optar por OpenStack, dado que tener un almacenamiento de objetos era un requisito clave y tenía una comunidad de usuarios en auge en la cual puedes resolver cualquier problema buscando en la documentación y/o en los foros de ayuda.

Se estuvo evaluando la posibilidad de desplegar la infraestructura en alguno de los servicios de computación en la nube ya existente, como por ejemplo **Amazon**, **Google**

# 1 - Introducción

---

**Cloud Engine o Microsoft Azure**, pero se descartó esta idea ya que, por una parte el BIFI dispone del hardware necesario y del personal técnico adecuado para mantener una infraestructura de este tipo. Por otra, el número de horas de uso de la infraestructura es muy elevado, lo cual no lo hacía económicamente viable.

No se consideraron alternativas que no fueran software libre, ya que el coste de las licencias dado el gran número de máquinas habría hecho que el coste económico para el Instituto y la Universidad de Zaragoza fuera inasumible.

## 1.6 Trabajo realizado por el autor

El trabajo realizado por el autor en este PFC para llegar al objetivo de desplegar una infraestructura de computación en la nube basada en OpenStack, totalmente funcional, ha sido el siguiente:

- Captura de los requisitos necesarios para migrar la infraestructura de computación distribuida basada en grid existente, a una infraestructura distribuida de computación en la nube basada en OpenStack.
- Primer despliegue de la nueva infraestructura de computación distribuida en la nube basada en OpenStack en base a los requisitos en la infraestructura de computación distribuida basada en grid que el BIFI administraba en las instalaciones de la Escuela Politécnica Superior de Huesca, (EPSH).
- Pruebas y validación del correcto funcionamiento de la infraestructura en base a los requisitos.
- Carencias y oportunidades de mejora para el futuro.
- Segundo despliegue de la nueva y mejorada infraestructura de computación distribuida en la nube basada en OpenStack.
- Pruebas y validación del correcto funcionamiento de la infraestructura en base a los requisitos.
- Carencias y oportunidades de mejora para el futuro.
- Seguimiento del correcto funcionamiento de la infraestructura.

## 1.7 Contenido de la documentación

Este documento consta de:

- **Memoria:** Pagina 1, que a su vez contiene los siguientes apartados:

# 1 - Introducción

---

- **Introducción:** el presente capítulo, en el que detallo el contexto, la motivación y los objetivos del proyecto.
- **Primer despliegue, EPSH:** Capítulo que describe el despliegue realizado en las máquinas de la Escuela Politécnica superior de Huesca. Decisiones de diseño, implementación y oportunidades de mejora.
- **Segundo despliegue, BIFI:** Capítulo que describe el segundo despliegue realizado en las máquinas de las que dispone el BIFI en Zaragoza, teniendo en cuenta además las oportunidades de mejora del primer despliegue. Decisiones de diseño, implementación y oportunidades de mejora.
- **Conclusiones:** Capítulo final con breves conclusiones, seguimiento de la infraestructura y valoraciones finales.

## 2 - ¿Qué es OpenStack?

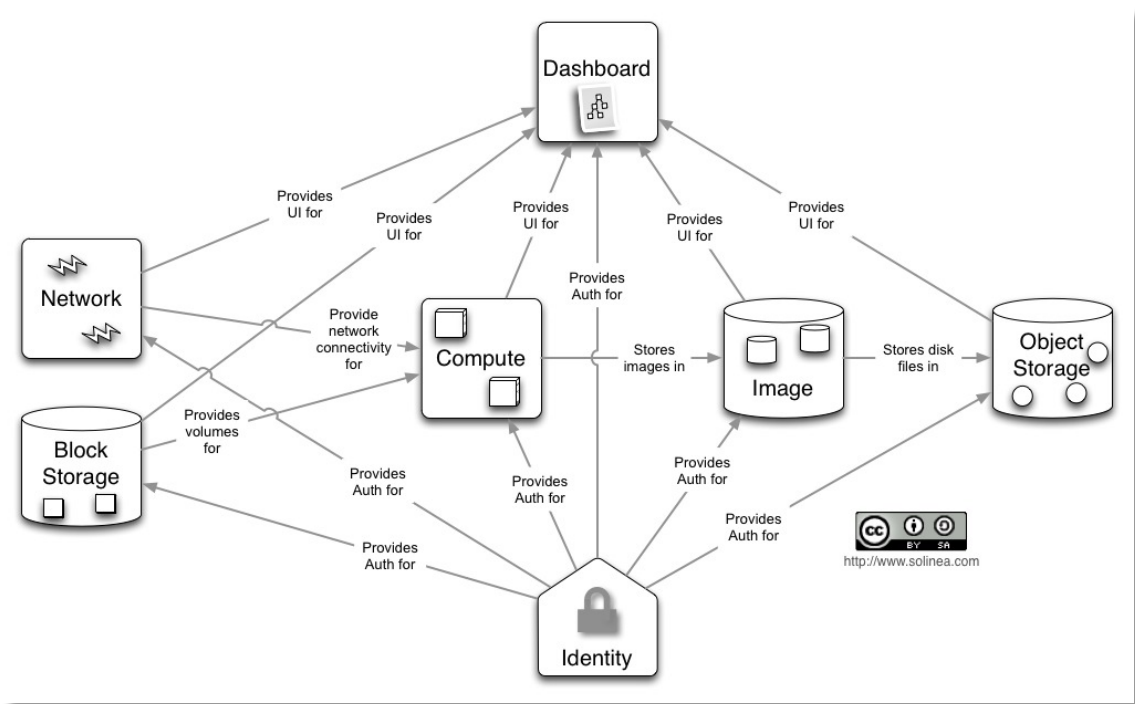
### 2.1 ¿Qué es OpenStack?

Para entender este PFC en su totalidad, hace falta entender primero que es y cómo funciona OpenStack.

OpenStack es un proyecto de software libre para proporcionar una infraestructura distribuida de computación en la nube, con dos objetivos destacados: ser capaz de funcionar sobre cualquier tipo de hardware y ser fácilmente escalable.

OpenStack consiste de una serie de servicios modulares relacionados entre sí, que son los encargados de proveer a la infraestructura de sus diferentes funcionalidades, almacenamiento, redes, autenticación entre otros.

OpenStack a día de hoy tiene un gran número de servicios con diferentes funcionalidades, pero después de hacer un análisis detallado de los mismos, se concluye que los servicios mínimos y las relaciones que debe de haber entre ellos, son los que se muestran en la **figura 1**:



*Figura 1 - Arquitectura básica de OpenStack (<http://www.solinea.com>)*

A continuación, detallo una pequeña descripción para cada servicio para y así una idea general de que hace cada uno de ellos:

- **Servicio de Identidad, Keystone (Identity en la figura):** Provee a la infraestructura de un directorio central de usuarios y la relación de los servicios a los que puede acceder cada usuario. Actúa como un sistema de autenticación común para el resto de servicios de OpenStack y puede ser integrado con otros

## 2 - ¿Qué es OpenStack?

---

servicios como LDAP. Adicionalmente, provee un listado de todos los servicios desplegados en la infraestructura.

- **Servicio de imágenes, Glance (Image en la figura) :** Provee a la infraestructura de las capacidades de descubrir, registrar y entregar imágenes de sistemas operativos, que serán utilizadas para instanciar las máquinas virtuales. Puede almacenar las imágenes en una gran cantidad de ubicaciones distintas, incluso en el almacenamiento de objetos de OpenStack (Swift) del que hablaremos más adelante.
- **Servicio de almacenamiento de bloques, Cinder (Block storage en la figura):** El servicio de almacenamiento de bloques, Cinder, provee a OpenStack de la capacidad de crear dispositivos persistentes de almacenamiento a nivel de bloque. Se encarga de la creación, destrucción, conexión y desconexión de dichos dispositivos de bloques a las máquinas virtuales. Además de poder utilizar el almacenamiento local del servidor, puede ser conectado a otros sistemas de almacenamiento, como Ceph o GlusterFS.
- **Servicio de cómputo, Nova (Compute en la figura):** La parte central de la infraestructura, diseñado para manejar y automatizar conjuntos de recursos de computación, capaz de trabajar con una amplia variedad de tecnologías de virtualización, como KVM, VMWare, Xen... Se encargará de crear las máquinas virtuales, de su correcto funcionamiento y su destrucción. Diseñado para ser altamente escalable.
- **Servicio de redes (Network en la figura):** En esta parte hay dos aproximaciones posibles, y las dos han sido utilizadas en este proyecto.  
Por una parte, existe el servicio denominado nova-network, que era el servicio más antiguo que existía y también el más estable que se utilizó para el primer despliegue.  
Posteriormente apareció Neutron, que le daba al usuario la total libertad de crear la topología de red que deseará en sus máquinas virtuales y que sería utilizado en el segundo despliegue.
- **Servicio de almacenamiento de objetos, Swift (Object storage en la figura):** Es un sistema de almacenamiento escalable y redundado. Los objetos y los ficheros son escritos en múltiples discos distribuidos en los servidores de la infraestructura, siendo OpenStack el encargado de asegurar la replicación y la integridad de los datos. Debido a esto, puede escalar de forma rápida y barata utilizando discos duros normales y corrientes.
- **Servicio de Dashboard, Horizon:** Provee a la infraestructura de una interfaz web para que los usuarios con menos experiencia puedan interactuar de una forma rápida y sencilla con la infraestructura.



# 3 - Herramientas utilizadas

---

## 3.1 Herramientas utilizadas

Las herramientas que se han utilizado para el PFC son las siguientes:

- **Plataforma de computación en la nube:** OpenStack (Grizzly, Kilo)
- **Sistema Operativo:** Ubuntu 12.04 LTS para todas las máquinas de la infraestructura de EPSH, Ubuntu 14.04 LTS para todas las máquinas de la infraestructura BIFI
- **Herramientas de gestión de la configuración y aprovisionamiento:** Ansible, Clonezilla
- **Herramientas de monitorización de la infraestructura:** Zabbix, Monit, Kibana + Logstash + Elasticsearch
- **Herramientas de copia de seguridad de la infraestructura:** Bareos
- **Herramienta de gestión de contenedores:** Docker
- **Sistema de almacenamiento para la infraestructura:** Ceph

¿Por qué se eligieron estas herramientas y no otras?

- **OpenStack Versión Grizzly para el primer despliegue y Versión Kilo para el segundo:** Grizzly era la última versión estable disponible en el momento de hacer el primer despliegue. En el caso del segundo despliegue, se decidió dar el salto a Kilo, dado que arreglaba numerosos bugs, añadía varios parches de seguridad y nuevas funcionalidades como detallaré más adelante.
- **Ubuntu 12.04 LTS y Ubuntu 14.04 LTS:** Ubuntu siempre ha ido de la mano de OpenStack, sus paquetes están en los repositorios de Ubuntu y siguen el mismo ciclo de actualizaciones de modo que fue una elección natural.

En el primer caso, se utilizó la versión 12.04 de Ubuntu ya que era la única versión **LTS – Long Term Support** - disponible hasta el momento. En el segundo despliegue se cambió a la más actual del momento, 14.04 debido a que era un requisito para instalar la última versión de OpenStack, Kilo.

Las versiones LTS de Ubuntu, son las versiones de Ubuntu más estables, debido a que no siempre tienen las últimas versiones de todo el software, pero a cambio son las más estables. Reciben actualizaciones críticas y de seguridad durante 5 años, lo cual las hace ideales para este tipo de despliegues.

- **Ansible y Clonezilla:** Ansible es una herramienta de software libre que permite configurar y administrar máquinas de una forma muy sencilla mediante “recetas”. En el caso de Ansible se eligió esta herramienta por ser muy sencillo de utilizar y requerir una instalación mínima. Otras herramientas que se probaron en una instalación de pruebas, como Puppet, tenían una curva de aprendizaje más elevada así como de una instalación más compleja. En el caso de Clonezilla, se eligió porque ya había una solución desarrollada en el BIFI basada en él para hacer los despliegues de la infraestructura de grid, y fue muy sencillo hacer los cambios para adaptarla a la nueva infraestructura de computación en nube privada.

# 3 - Herramientas utilizadas

---

- **Zabbix, Monit y Kibana + Logstash + Elasticsearch:** En el caso de Zabbix Se evaluaron y probaron otras plataformas de monitorización como Nagios, pero se eligió Zabbix debido a su sencillez para instalarlo/ actualizarlo, a que es mucho más completo ya que prácticamente sin hacer ningún cambio monitoriza un gran abanico de aspectos básicos y permite definir de una forma sencilla chequeos personalizados utilizando Python o cualquier otro lenguaje.

Adicionalmente a Zabbix, se instaló una solución para almacenar y recopilar los registros de eventos utilizando Logstash + Kibana + Elasticsearch. Dicha solución permite mediante una interfaz web y de forma rápida y sencilla hacer búsquedas en todos los logs que se quiera para buscar posibles intrusiones, fallos, etcétera.

También se hicieron pruebas con rsyslog, pero finalmente se optó por Logstash + Kibana + Elasticsearch por ser esta opción mucho más potente y amigable al usuario final.

Por último, se utilizó Monit cuya finalidad era asegurar que ciertos servicios críticos estaban funcionando dado que el autor ya tenía experiencia trabajando con el anteriormente.

- **Bareos:** Para hacer las copias de seguridad de la infraestructura, se utilizó Bareos, que automatiza todo lo referente a las copias de seguridad: Hace las copias en un horario definido, y avisa del resultado de las copias por email a los administradores de la infraestructura

Se hicieron pruebas tanto con Bacula como con Bareos, pero finalmente se eligió a Bareos, por ser un fork totalmente libre de Bacula, con una comunidad de usuarios más amplia y con actualizaciones más frecuentes, tanto de seguridad, como de nuevas funcionalidades. En Bacula, al existir una versión de pago, hay algunas actualizaciones que no llegan jamás a la versión libre.

- **Docker:** Docker, es un proyecto de software libre, que permite automatizar el despliegue de una aplicación dentro de un contenedor, proporcionando de una capa de abstracción y automatización.  
Dicho de otra manera, permite empaquetar una aplicación y todas sus dependencias en un contenedor que se puede ejecutar en cualquier servidor Linux, lo cual nos da una gran flexibilidad y portabilidad.
- **Ceph:** Ceph se utilizó como sistema de almacenamiento de objetos de la infraestructura, ya que es sencillo de instalar y se ha convertido en un estándar “*de facto*” en OpenStack, no solo para la parte de objetos.

La otra opción considerada fue instalar el almacenamiento de objetos con la solución de OpenStack, Swift, pero se descartó por su curva de aprendizaje

# 4 - Primer Despliegue (EPSH)

## 4.1 Objetivos

El objetivo primordial de este primer despliegue, era migrar la infraestructura de grid existente en EPSH y que contaba con pocos usuarios, a una infraestructura de computación en nube privada basada en OpenStack, con el objetivo de familiarizarse con el nuevo concepto de nube privada, y probar las funcionalidades con un grupo reducido de usuarios antes de dar el paso a un grupo más amplio y numeroso.

### 4.1.1 Restricciones de la infraestructura preexistente

Antes de nada, hay que tener en cuenta que la infraestructura se desplegó sobre un hardware que ya existía y que no estaba pensando específicamente para este uso, lo cual ocasionaba que tuviéramos una serie de restricciones que son las siguientes:

- Los nodos de cómputo de la infraestructura tienen únicamente un disco.
- La infraestructura solo dispone de un SAI y una acometida eléctrica.
- Había 4 nodos singulares que se utilizaban como nodos de servicio de la infraestructura grid, con más RAM y almacenamiento redundante en modo RAID 1 por hardware y 36 nodos de cómputo, que no tenían almacenamiento redundante RAID y tenían menos memoria disponible.
- Además de esto, había una cabina de discos dedicada al almacenamiento. Dicha cabina, consta de varios discos con los cuales se pueden crear unidades lógicas de distinto tamaño y distinto nivel de redundancia utilizado para ello la controladora RAID que lleva integrada. Dichas unidades se puede utilizar posteriormente en cualquier máquina de la infraestructura utilizando ISCSI.

Estas mismas restricciones se aplican en el segundo despliegue posteriormente, ya que el hardware es idéntico.

### 4.1.2 Captura de requisitos mínimos

Con estas restricciones en mente, se realiza el análisis y se capturan los requisitos que debe tener la infraestructura en el **cuadro 1**.

Por una parte están los requisitos básicos, que son los necesarios para el correcto funcionamiento de la infraestructura, y los deseables, que confieren más calidad a la misma. También se distingue entre requisitos funcionales y no funcionales. Por último, se detalla también si al final del proyecto se ha conseguido cumplir con dichos objetivos.

Requisito	Descripción	Tipo	Finalización
R1	La infraestructura funcionara en su totalidad en sobre Ubuntu 12.04	B – NF	Sí

## 4 - Primer Despliegue (EPSH)

R2	La versión de OpenStack que utilizará la infraestructura, será Grizzly	B – NF	Sí
R3	Todas las máquinas de la infraestructura, deberán tener salida a Internet y una IP pública para acceder a ellas	B - NF	Sí
R4	Se deberá poder asignar dinámicamente una IP pública a cada máquina virtual que se instancie en la infraestructura	B - NF	Sí
R5	La infraestructura dispondrá de un servidor de bases de datos para todas las bases de datos requeridas por los diferentes servicios	B - NF	Sí
R6	La infraestructura dispondrá de un espacio de almacenamiento redundado para almacenar las imágenes del servicio Glance	B - NF	Sí
R7	La infraestructura dispondrá de un espacio de almacenamiento redundado para proveer almacenamiento a nivel de bloque	B - NF	Sí
R8	El almacenamiento a nivel de bloque, utilizará LVM para la creación, manejo y destrucción de los dispositivos de bloques	D - NF	Sí
R9	La infraestructura dispondrá de un espacio de almacenamiento sin redundancia para el servicio de almacenamiento de objetos, Swift	B - NF	Sí
R10	Todas las comunicaciones entre servicios, así como la interfaz web Horizon, utilizarán un canal seguro de comunicación (SSL), o en su defecto, utilizarán la red interna	D - NF	Sí
R11	Toda la infraestructura estará monitorizada en todo momento, tanto a nivel de hardware, como a nivel de servicios	B - NF	Sí
R12	El sistema de monitorización de la infraestructura, será capaz de alertar a los administradores en caso de detectar cualquier anomalía	B -NF	Sí
R13	La infraestructura deberá contar con un sistema de copias de seguridad, que se encargará al menos de hacer copia de seguridad de la configuración de la infraestructura	B - NF	Sí
R14	Los administradores de la infraestructura solo deberán encargarse de la administración de la misma	B - NF	Sí
R15	La infraestructura no deberá tener un punto único de fallo	D - NF	No

*Cuadro 1 - Cuadro de requisitos de la infraestructura (EPSH)*

Unos breves comentarios sobre algunos requisitos:

## 4 - Primer Despliegue (EPSH)

---

- **R4:** Para cumplir con este requisito, se decidió utilizar nova-network, ya que era la opción más madura y estable a la hora de desarrollar este primer despliegue y que permitía cumplir este requisito.
- **R6 y R7:** Dichos servicios tienen que estar sobre un almacenamiento redundado para no perder datos de los usuarios en caso de un fallo en algún disco. Dado que el espacio que necesitan además estos servicios pueden ser del orden de terabytes, se decidió crear una unidad lógica para cada servicio en la cabina de discos, para disponer de un gran espacio de almacenamiento y con un nivel básico de redundancia al estar sobre un RAID.
- **R8:** La decisión de utilizar LVM para gestionar los dispositivos de bloques viene dada porque era la única opción viable con el hardware disponible. Se estudiaron y se hicieron pruebas con otros sistemas de almacenamiento como Ceph o GlusterFS para los dispositivos de bloques de OpenStack, pero se descartaron finalmente ya que la red de 1 Gigabit era un cuello de botella para todos ellos.
- **R9:** En el caso de Swift, no es necesario que vaya redundado el espacio, ya que el propio servicio es el que se encarga de proveer esta redundancia. Para ello, lo que se hizo fue partir el espacio destinado para Swift en dos volúmenes distintos, y que fuera el propio Swift el encargado de gestionarlos.

### 4.1.3 Casos de uso

Las diferentes funcionalidades que podrá realizar un usuario una vez desplegada la infraestructura, son las siguientes:

- Crear / Destruir máquina virtual
- Añadir / Eliminar dispositivo de bloques
- Subir / Descargar ficheros del almacenamiento de objetos
- Obtener estadísticas de uso de la infraestructura

Todo esto lo podrán hacer desde una sencilla interfaz web destinada a tal efecto.

### 4.2 Diseño y planificación temporal

La planificación temporal que se realizó fue la que podemos observar en la **figura 2**.

En dicha figura podemos ver las diferentes tareas que se llevaron a cabo para tener toda la infraestructura desplegada y en producción. Además, podemos ver dos hitos importantes: el primero, la entrada en producción de dicha infraestructura en la nube privada en junio del 2013, y posteriormente, teniendo en cuenta el éxito que tuvo la infraestructura con los usuarios del BIFI, su posterior integración con el proyecto europeo de **EGI-Engage** en Diciembre del 2013.

# 4 - Primer Despliegue (EPSH)

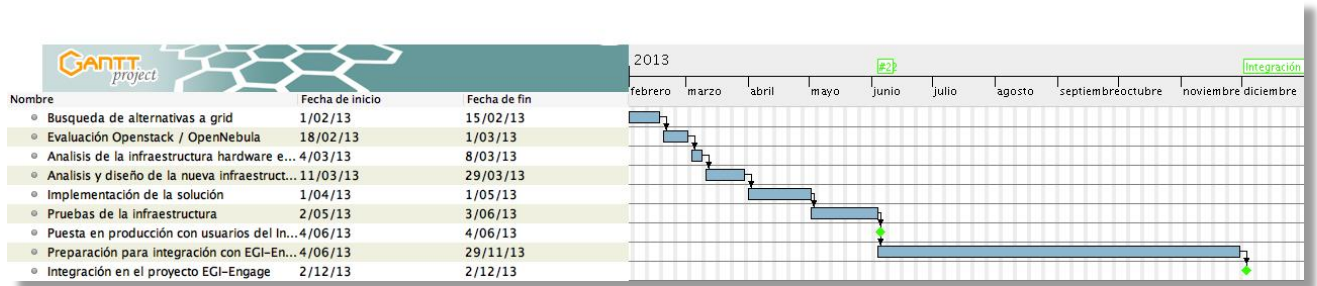


Figura 2 - Planificación temporal (EPSH)

No hubo retrasos en dicha planificación, ya que se reservó un tiempo más que suficiente para el desarrollo de las diferentes tareas.

Respecto a la última tarea, la integración con el proyecto **EGI-Engage** su duración es mucho más alargada que las demás por toda la burocracia asociada a los proyectos europeos, más que por la implementación técnica en sí.

Una vez realizado el análisis, habiendo recopilado todas las funcionalidades que ha de tener la nueva infraestructura, y recopilados todos los requisitos, se pasa a la fase de diseño. En este punto, decidí que máquinas iban a actuar como servicios centrales de la infraestructura, y que máquinas iban a actuar como servidores de cómputo albergando únicamente máquinas virtuales. Así pues, vamos a diferenciar entre nodos de servicio y nodos de cómputo.

## 4.2.1 Nodos de servicio

La infraestructura, como ya he mencionado anteriormente consta de 4 nodos de servicio. Todos ellos disponen de 4 interfaces de red que les permiten conectarse: a la cabina de discos mediante el protocolo iSCSI, a internet mediante una IP pública, entre ellos y el resto de nodos de cómputo formando una red privada, y por último una red para la gestión mediante IPMI.

En este punto de la instalación no hizo falta modificar el cableado a nivel de red, ni alterar la configuración de los switches de red a nivel lógico.

- **Server1-epsh:** Servidor de almacenamiento especializado en almacenamiento de imágenes y dispositivos de bloques. Dispone de dos dispositivos de bloques de la cabina de discos, uno de ellos utilizado para almacenar las imágenes que utilizará OpenStack para crear las máquinas virtuales, y el otro de ellos que será utilizado por Cinder, para todo lo relativo al almacenamiento de bloques de la infraestructura de OpenStack.

Se ha utilizado un solo servidor para estos dos servicios, ya que si bien son operaciones pesadas que pueden necesitar de un gran ancho de banda – por ejemplo subir a Glance una nueva imagen – son operaciones que no son muy frecuentes en el uso habitual de la infraestructura.

## 4 - Primer Despliegue (EPSH)

---

- **Server2-epsh:** Servidor de almacenamiento especializado en el almacenamiento de objetos. Dispone de un dispositivo de bloques para Ceph, que será el encargado de dar el servicio de almacenamiento de objetos a la infraestructura.

La razón de utilizar otro servidor para el almacenamiento es puramente técnica: de este modo, podemos utilizar otra de las interfaces de red de la cabina de discos para dedicarla única y exclusivamente al tráfico del almacenamiento de objetos, ya que en principio se prevé que conforme el número de usuarios, el número de peticiones a este servicio aumente.

- **Server3-epsh:** Servidor de interconexión. Conecta con la VPN que había definida previamente en el BIFI para permitir el acceso remoto a los servidores, tanto a nivel de IPMI, como a la ILO, una interfaz de gestión web propietaria de HP. Además de la VPN reporta al servidor central de Zabbix los datos de toda la infraestructura.
- **Server4-epsh:** Servidor controlador de la infraestructura. Posee el servidor de bases de datos MySQL que almacena las bases de datos del resto de servicios, RabbitMQ Server, como servidor de paso de mensajes, requerido por varios de los servicios de OpenStack, Keystone, servicio de autenticación de la infraestructura, y nova-api, API de la parte de computo de la infraestructura, así como de otros servicios menores relacionados con la API.

### 4.2.2 Nodos de cómputo

La parte de los nodos de cómputo es mucho más breve. Disponen de dos interfaces de red físicas, más una interfaz de red adicional virtual. Los detalles de porque se hizo de esta manera los detallaré más adelante en el capítulo de implementación.

Todos los nodos de cómputo son análogos y poseen la misma configuración, en la figura solo he representado uno de ellos por sencillez.

### 4.2.3 Almacenamiento

El almacenamiento disponible para la infraestructura consiste en una cabina de discos que permite crear unidades lógicas de diferente tamaño y niveles de RAID.

## 4.3 Implementación

La implementación de la infraestructura quedo de la siguiente manera:

- **Almacenamiento:** Para el almacenamiento de los diferentes servicios de OpenStack, como Cinder y Glance, se reutilizó la cabina de discos creando las diferentes unidades lógicas que luego serían exportadas por iSCSI como he detallado en la parte de “Nodos de Servicio”. Además de este almacenamiento para estos dos servicios concretos, los 4 nodos singulares de servicio disponen de un RAID 1 o RAID en espejo de 250GB que los hacen tolerantes a un fallo de disco. Los nodos de computo no disponen de ningún tipo de RAID y únicamente

## 4 - Primer Despliegue (EPSH)

---

tienen un disco duro SATA de 500GB donde almacenar las máquinas virtuales que se crean en ellos

- **Redes:** Para explicar las redes, hace falta entender como agrupa OpenStack a los usuarios.

Para trabajar en OpenStack, un usuario debe estar dentro de un proyecto. Dicho proyecto, tendrá una red privada para interconectar todas las máquinas virtuales de dicho proyecto y opcionalmente, a cada máquina virtual se le podrá asignar una IP pública de un conjunto de IP's configurado globalmente para toda la infraestructura, siempre y cuando ese proyecto no haya agotado su cuota de IP's públicas.

El problema viene con las redes privadas de cada proyecto, ya que si no se aíslan de alguna manera, el usuario del proyecto A, podría ver de manera relativamente sencilla el tráfico de las máquinas virtuales del proyecto B.

Dado que no es una característica deseada que los usuarios de un proyecto sean capaces de ver el tráfico de red de los usuarios de otros proyectos, optamos por asignar a la red de cada proyecto dentro de OpenStack su propia VLAN, de modo que cada usuario solo fuera capaz de ver el tráfico dirigido a sus máquinas virtuales. Para ello se definieron las VLANES necesarias en el equipamiento de red de la infraestructura

Aparte de esto, la configuración de nova-network se hizo utilizando el modo "*multi-host*" que permite que cada nodo de cómputo tenga su propio nova-network, de modo que si cae el servicio nova-network de uno de los nodos, no afecte a la conectividad del resto de nodos de cómputo.

- **Monitorización:** La monitorización de la infraestructura se centralizó en un Zabbix que está ubicado en el CPD del BIFI. Esto se hizo así para en caso de una caída completa de la infraestructura, ser conscientes de lo que ha sucedido y que no caiga la monitorización si cae también la infraestructura.
- **Copias de seguridad:** Por una parte, se hacen copias de seguridad de la base de datos de la infraestructura, y por otra parte, Bareos hace automáticamente copias de seguridad de los ficheros de configuración críticos de los nodos de servicio de la infraestructura, para en caso de fallo de uno de ellos, poder recuperarlo rápidamente.

Se tomo la decisión de hacer copia de seguridad de la base de datos y de los ficheros de configuración críticos, porque es lo mínimo necesario e imprescindible para recuperar la infraestructura en caso de fallo de alguno de las máquinas de servicio.

Una vez solventados estos aspectos, el método que se empleó para desplegar al completo la infraestructura fue el siguiente:

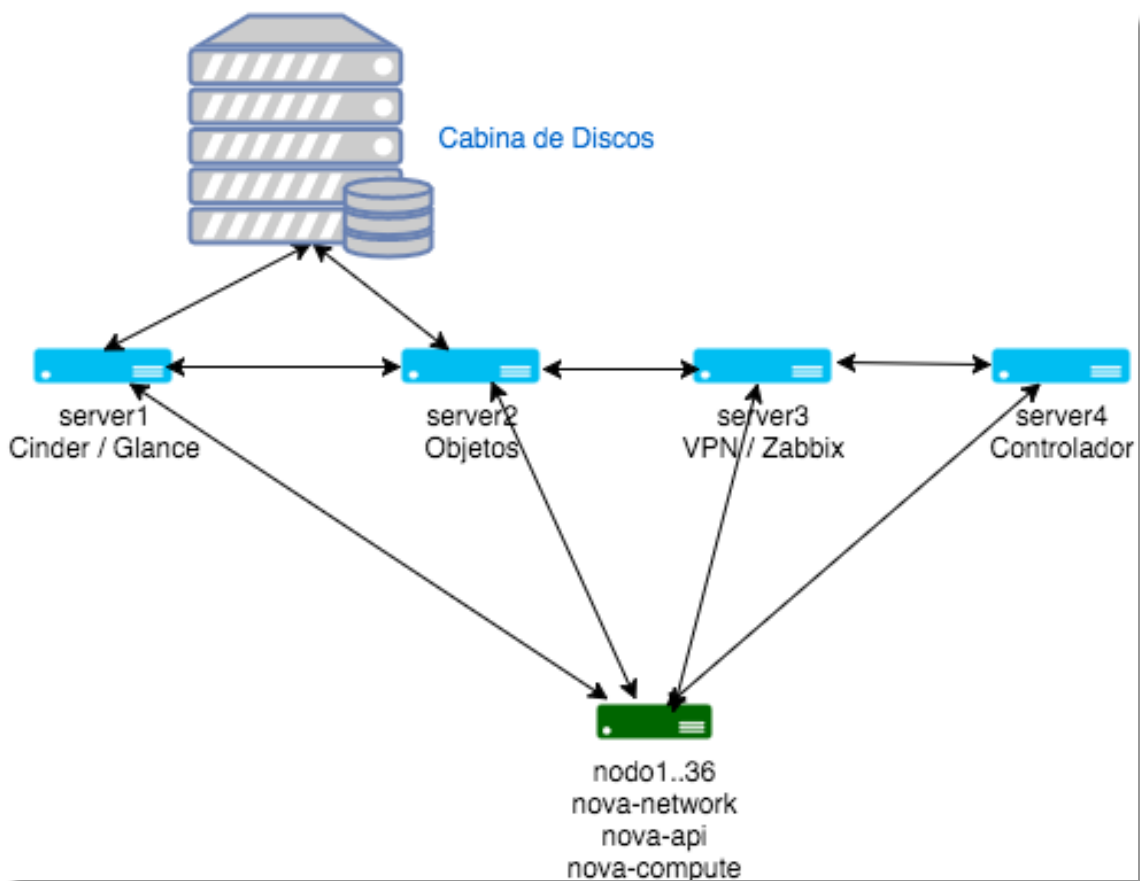


## 4 - Primer Despliegue (EPSH)

1. Se hizo la instalación de un nodo de servicio y de un nodo de cómputo. Una vez hecho esto, se capturo con Clonezilla una imagen para cada tipo de nodo, que luego sería desplegada remotamente en cada uno de ellos dependiendo del tipo que fuere, utilizando PXE.
2. Una vez instalado el sistema operativo en todos los servidores, se procedió a instalar en cada servidor, siguiendo el esquema descrito en la figura 3 cada uno de los servicios que correspondían en cada máquina. Para ello, se desarrollaron unas “recetas” utilizando Ansible que se encargaron de automatizar la tarea para desplegar todo de una forma rápida y sencilla.
3. Se comprobó el buen funcionamiento de la infraestructura desplegada.

### 4.4 Diagrama de la infraestructura

De modo esquemático, la **figura 3**, representa como quedo la infraestructura una vez terminado el despliegue:



*Figura 3 - Esquema de la infraestructura (EPSH)*

Solo se ha representado un nodo de computo de la infraestructura por sencillez. Como se puede ver, están todos conectados con todos, gracias a la red interna de la que disponen.

## 4 - Primer Despliegue (EPSH)

---

Respecto a los servidores de almacenamiento, solo server1 y server2 conectan con la cabina de discos, ya que son los únicos que necesitan acceso directo a la misma.

Tampoco esta dibujado en el diagrama el acceso a internet para mantener la sencillez del mismo, pero se da por hecho, que todas las maquinas que en el aparecen, tienen acceso directo a internet.

### 4.5 Pruebas y validación

Se llevaron a cabo dos tipos de pruebas para comprobar el correcto funcionamiento de la infraestructura:

- El primer conjunto de pruebas, consistió en hacer pruebas de un caso de uso típico de la infraestructura: crear una máquina virtual, crear un dispositivo de bloques y añadirlo a la instancia, subir y bajar ficheros al almacenamiento de objetos, desconectar el dispositivo de bloques y destruir la instancia.

En este conjunto de pruebas no se detectó ningún tipo de problema en ninguna de estas acciones y tampoco posteriormente una vez que la infraestructura estaba en uso.

- El segundo conjunto de pruebas consistió en hacer pruebas de carga a la infraestructura para comprobar su comportamiento ante un gran número de usuarios. Se lanzaron de forma simultánea un gran número instancias para simular el comportamiento de un gran número de usuarios haciendo peticiones a la infraestructura simultáneamente, con el fin de comprobar el aguante de la misma. Se monitorizó la infraestructura con Zabbix mientras se realizaba. Más allá de un aumento en los niveles de carga de la infraestructura – CPU, memoria RAM utilizada, disco, etcétera -, no se detectó ningún problema en la prueba y todo funcionó según lo previsto.

### 4.6 Problemas detectados

Los problemas detectados en esta infraestructura fueron varios:

- **La infraestructura posee un punto único de fallo:** Esto es debido a varios motivos:
  - Solo hay un SAI el cual está alimentado por una acometida eléctrica, un fallo prolongado en cualquiera de los dos hará que falle la infraestructura.

La acción tomada fue informar a mi director de proyecto para tenerlo en cuenta y que pueda ser solventado en un futuro añadiendo un grupo electrógeno y añadiendo una nueva acometida.

## 4 - Primer Despliegue (EPSH)

---

- La base de datos no se encuentra replicada en la infraestructura utilizando algún tipo de clusterización, como por ejemplo MySQL Galera. Se estudió la posibilidad, pero finalmente se descartó por los problemas inherentes al caso concreto de la Universidad.

En la universidad, cada mes de agosto se hace una parada para el mantenimiento de las líneas eléctricas que alimentan los diferentes campus, lo cual nos obliga forzosamente a parar todas nuestras máquinas. En el caso de desplegar una base de datos utilizando algún tipo de clúster como MySQL Galera, haría que el clúster quedaría en un estado inconsistente puesto que Galera no es capaz de recuperar el clúster después de un apagón, porque asume que hay siempre al menos un miembro del clúster en funcionamiento.

Una posible solución que se está estudiando aun a día de hoy, es crear otra instancia de MySQL Galera fuera de la infraestructura para evitar este tipo de problemas

- Los servicios de la infraestructura no están redundados. Este problema se ha tenido en cuenta y se ha intentado solventar en la medida de lo posible en el segundo despliegue en el BIFI.
- **Los nodos de cómputo no disponen de tolerancia ante un fallo de disco:** si falla el disco de un nodo, todas las instancias se pierden irremediablemente. Esto es una restricción como hemos comentado antes por el hardware heredado.

Se anima encarecidamente a los usuarios a hacer copias de seguridad de sus instancias si tienen algún dato importante en ellas, y sobre todo a utilizar los servicios de Cinder y Swift para el almacenamiento los cuales disponen de redundancia.

### 4.7 Oportunidades de mejora

A la vista de los problemas detectados se pueden destacar varias oportunidades de mejora:

- Redundar los servicios de la infraestructura.
- Replicar la base de datos en un servidor adicional, utilizando algún esquema Maestro-Esclavo.
- Añadir un disco duro adicional a los nodos de computo para proveer de redundancia a las máquinas virtuales de la infraestructura.

Estas oportunidades de mejora fueron tenidas en cuenta a la hora de hacer el segundo despliegue.

# 4 - Primer Despliegue (EPSH)

---

## 4.8 Conclusiones

Este primer despliegue sirvió como un banco de pruebas perfecto para el segundo despliegue que realizaríamos posteriormente. Se dio servicio principalmente a los investigadores del BIFI, los cuales confirmaron la robustez prevista de la infraestructura desplegada.

Asimismo, la tarea de los administradores se redujo drásticamente, ya que ahora los usuarios eran responsables de sus propias máquinas y de lo que instalaban en ellas, logrando con este despliegue el objetivo que estábamos buscando desde un principio.

De cara a los usuarios menos expertos, esta infraestructura les facilitó los despliegues y configuraciones de sus aplicaciones en la misma, respecto de la anterior infraestructura cuyo manejo era mucho más complejo.

Por parte del autor, se adquirió una gran experiencia en el manejo de las diferentes herramientas utilizadas para este despliegue, experiencia que posteriormente simplificaría enormemente el segundo despliegue realizado.

# 5 - Segundo Despliegue (BIFI)

## 5.1 Objetivos

El objetivo principal de este despliegue viendo el éxito obtenido en el primero, era migrar el resto de infraestructura distribuida basada en grid a una infraestructura distribuida en la nube privada basada en OpenStack.

Para ello, el autor utilizó toda la experiencia adquirida en el primer despliegue, se reaprovechó gran parte de los scripts y archivos de configuración del primer despliegue, y se tuvieron en cuenta las oportunidades de mejora del despliegue anterior.

### 5.1.1 Captura de requisitos mínimos

Una vez realizado el primer despliegue en EPSH, realizadas las pruebas pertinentes para comprobar su funcionamiento, y habiéndolo lanzado en producción, se hizo un segundo despliegue en el CPD del BIFI (Zaragoza) de otra infraestructura de computación distribuida en nube privada basada en OpenStack, debido a la creciente demanda que había de recursos computacionales en la nube.

En términos de hardware, este nuevo despliegue era idéntico al anterior, lo que permitió reaprovechar de forma sencilla gran parte de lo ya desarrollado y aprendido. En cualquier caso, el análisis y captura de requisitos de esta nueva infraestructura, se hizo teniendo en cuenta las oportunidades de mejora detectadas.

En la tabla de requisitos que aparece en el **cuadro 3.1**, se han obviado los requisitos que son comunes con la anterior. Se han añadido un par de requisitos nuevos y se han tenido en cuenta las oportunidades de mejora descritas en el despliegue anterior:

<b>Requisito</b>	<b>Descripción</b>	<b>Tipo</b>	<b>Finalización</b>
R1	Toda la infraestructura deberá funcionar sobre Ubuntu 14.04	B – NF	Sí
R2	La versión de OpenStack que utilizará la infraestructura, será OpenStack Kilo	B - NF	Sí
R3	Los usuarios deben de ser capaces de crear cualquier topología de red para interconectar sus máquinas virtuales	B - NF	Sí
R4	La migración deberá estar hacerse durante el parón de verano	B - NF	Sí
R5	Todos los servicios tendrán que estar redundados en la infraestructura	D - NF	Parcialmente

# 5 - Segundo Despliegue (BIFI)

## Cuadro 2 - Cuadro de requisitos de la infraestructura (BIFI)

Algunos comentarios sobre los requisitos:

- **R3:** En este caso, dado que Neutron había avanzado y ya era un proyecto estable y maduro, se decidió que era apropiado desplegarlo ya que era un requisito indispensable que los usuarios pudieran manejar también sus propias redes en las máquinas virtuales, de cara a simplificar la tarea del administrador de la infraestructura.
- **R4:** La migración debía hacerse en el parón de verano de la Universidad y debía estar completada antes del termino de dicho parón, para afectar lo mínimo a los usuarios. Se advirtió con tiempo a los usuarios de la migración del servicio de grid a un servicio de computación en la nube y se avisó con tiempo suficiente antes de dicho parón para que recuperaran sus datos de la infraestructura de grid existente.
- **R5:** Era un requisito deseable, pero no imprescindible, conseguir que todos los servicios estuvieran redundados para evitar posibles fallos en la infraestructura en caso de que una de las máquinas de servicios fallará. Esto fue conseguido en parte como detalle posteriormente.

### 5.1.2 Casos de uso

Al caso de uso típico, se le añade ahora la posibilidad de que el usuario se configure cualquier tipo de red en sus máquinas virtuales.

## 5.2 Diseño y planificación temporal

La planificación temporal que se realizó fue la que podemos observar en la **figura 4**:

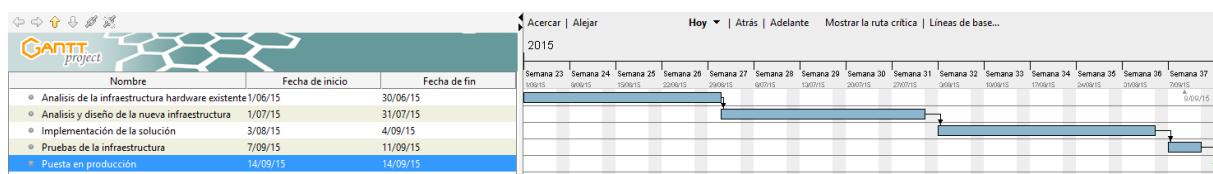


Figura 4 - Planificación temporal (BIFI)

En este nuevo despliegue, se utilizaron muchos más días en las tareas de análisis, ya que al ser verano la dedicación fue a tiempo parcial.

En el despliegue de la solución se trabajó a tiempo completo, ya que el objetivo era cumplir con las fechas y que la infraestructura estuviera probada y en producción una vez terminará el parón de verano.

Toda la planificación se cumplió sin mayores incidencias.

El diseño de la infraestructura cambia ligeramente, ya que se decidió utilizar el servicio elegido para la creación de las redes virtuales fue Neutron.

# 5 - Segundo Despliegue (BIFI)

---

Además de este cambio en las redes, ahora hay una máquina que hace de proxy y reparte el tráfico de los distintos servicios, con el objetivo de equilibrar la carga cuando muchos usuarios utilizan la plataforma simultáneamente, y para conseguir en la medida de lo posible redundar el máximo número de servicios posibles. La idea principal es que cada máquina tenga al menos 1 copia de los servicios esenciales de la infraestructura, entendiendo por servicio esencial de la infraestructura los siguientes: **Keystone, Glance, Cinder, Neutron, Nova-API**. Con alguno de ellos, ha sido posible redundarlos, y con otros no. En el capítulo de implementación de esta infraestructura hablaré más sobre este tema.

## 5.2.1 Nodos de servicio

En este punto no hay nada que resaltar respecto a los nodos de servicio que se utilizaron en la primera infraestructura, ya que son idénticos, a nivel de hardware.

En este punto de la instalación tampoco hizo falta modificar el cableado a nivel de red, ni alterar la configuración de los switches de red a nivel lógico.

- **Server1-bifi:** Servidor de almacenamiento. Análogo a server1-epsh en el primer despliegue.
- **Server2-bifi:** Servidor de almacenamiento de objetos. Análogo a server2-epsh.
- **Server3-bifi:** Servidor de interconexión. Análogo a server3-epsh. Sin embargo, se ha instalado Neutron en este servidor, que es el nuevo servicio para la red de OpenStack.  
Neutron es el nuevo servicio que se encarga de controlar las redes virtuales en OpenStack y debido a la carga que lleva asociada a dicho servicio, se ha decidido dedicar un único servidor a esta tarea.
- **Server4-bifi:** Servidor controlador de la infraestructura. Análogo a server4-epsh, salvo que en este caso posee además un Haproxy para hacer de nodo cabecera de la infraestructura, el cual será el encargado de repartir las peticiones de los diferentes servicios a los distintos servidores.

En todos ellos se decidió instalar Docker. El motivo de esta decisión, es el siguiente: Ahora todos los servidores van a tener un contenedor de cada uno de los siguientes servicios

- Keystone
- Nova (API)
- Glance (API)
- Cinder (API)
- Neutron (API)

La idea es redundar al menos esta parte de cada uno de los servicios. Desgraciadamente, el grueso de otros servicios no se ha podido redundar por toda la complejidad inherente a ellos. Es el caso por ejemplo de Neutron, que hace uso de los espacios de nombres de

# 5 - Segundo Despliegue (BIFI)

---

Linux para la creación de las redes virtuales, y necesita acceder directamente al servidor, suponiendo esto un punto único de fallo del que hablaré más adelante.

Una vez que se dispone de un contenedor lanzado en cada uno de los servidores utilizando Docker, se puede balancear la carga entre todos ellos y se puede proveer de una redundancia básica en al menos los servicios descritos en la página anterior.

## 5.2.2 Nodos de cómputo

La parte de los nodos de cómputo en esta infraestructura es análoga a los servidores de cómputo de EPSH ya que poseen el mismo hardware.

En la parte de servicios, destacar que desaparecen nova-api y nova-network, y son sustituidos por Neutron, que serán el encargado de manejar las redes virtuales.

## 5.2.3 Almacenamiento

Análogo al almacenamiento existente en EPSH

## 5.3 Implementación

Este despliegue es totalmente análogo al de EPSH salvo por la parte de del servicio encargado de las redes virtuales.

En este nuevo despliegue se utilizó Neutron como servicio para crear las redes de las máquinas virtuales, ya que había madurado desde el último despliegue y ofrecía características adicionales como permitir cualquier tipo de topología de red.

Adicionalmente, simplifica la parte de administración de la infraestructura, ya que Neutron utiliza por debajo la tecnología **GRE** – *Generic Routing Encapsulation* - para el manejo de las redes virtuales de la infraestructura, lo que hace que no sea necesario definir todas las VLANES en el equipamiento de red como era necesario hacer con nova-network

La desventaja de Neutron es que ahora todo el tráfico de red de las máquinas virtuales pasa por server3-bifi, de modo que si cae este servidor, todas las máquinas virtuales de la infraestructura pierden la conectividad. Esto es así hasta el día de hoy debido a el diseño de Neutron y fue tenido en cuenta a la hora de hacer este despliegue.

Por otra parte, ya he comentado la existencia de un proxy que reparte equilibra las peticiones de los diferentes servicios, a cada uno de los contenedores desplegados en la infraestructura.

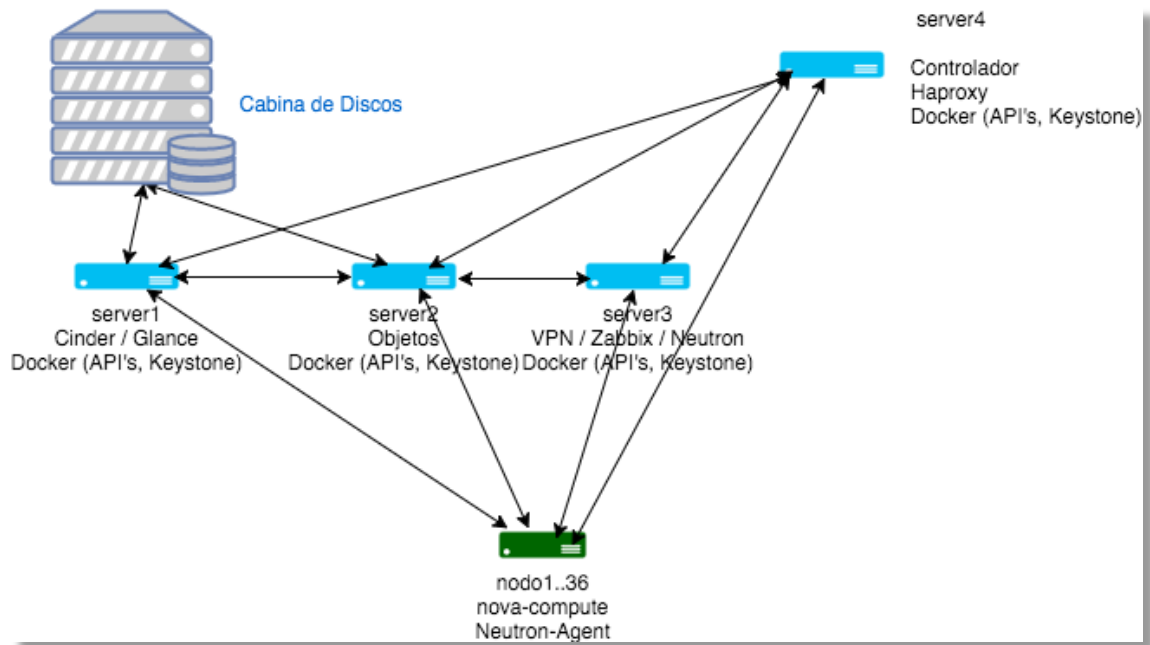
A la hora de desplegar la infraestructura se aprovechó gran parte de lo ya aprendido anteriormente y se reutilizaron gran parte de los scripts y archivos de configuración de la primera infraestructura con algunos cambios mínimos para adaptarlos a la nueva.



# 5 - Segundo Despliegue (BIFI)

## 5.4 Diagrama de la infraestructura

De modo esquemático, la **figura 5**, representa como quedo la infraestructura una vez terminado el despliegue:



*Figura 5 - Esquema de la infraestructura (BIFI)*

En este caso, la diferencia mas importante es la de server4 actuando como proxy. Recibe las peticiones para los diferentes servicios y él se encarga de distribuirlas a los contenedores situados dentro de el mismo, así como en el resto de los servidores.

No aparece representado en la figura por sencillez, pero una vez más se supone que todas las máquinas en la infraestructura disponen de salida a internet.

Tampoco aparece por sencillez, pero todo el tráfico de las máquinas virtuales, pasa ahora por server3.

## 5.5 Pruebas y validación

Las pruebas realizadas fueron las mismas que en la infraestructura de EPSH. No se detectó ningún problema en ninguna de ellas.

## 5.6 Problemas detectados

Los problemas detectados en la infraestructura son los siguientes:

- **La infraestructura tiene un punto único de fallo:** Por varios motivos
  - Eléctricamente, análogo a EPSH, un solo SAI y una sola línea eléctrica.

## 5 - Segundo Despliegue (BIFI)

---

- Todo el tráfico de las máquinas virtuales pasa ahora por server3-bifi por restricciones en el diseño de Neutron. Se evaluaron algunas soluciones para duplicar dicho servidor con **Pacemaker** / **Corosync**, - herramientas específicamente pensadas para tal uso y que he dejado fuera de esta memoria - pero fueron descartadas por mi director de proyecto, ya que no disponían del hardware suficiente.
- Seguimos teniendo como punto de fallo la MySQL por lo comentado en EPSH.
- Si bien hemos redundado los servicios, no tenemos el proxy que conecta con todos ellos redundando. Había que redundar este proxy utilizando alguna opción tipo **Pacemaker** / **Corosync** pero una vez más se descarto por falta de hardware suficiente.

- **Los nodos de computo siguen teniendo un solo disco duro**

### 5.7 Oportunidades de mejora

A la vista de los problemas detectados, se pueden destacar aquí también varias oportunidades:

- Replicar la base de datos en un esquema maestro esclavo, así como el resto de servicios problemáticos, como Neutron, Cinder y el proxy de la infraestructura.
- Añadir un disco duro a los nodos de cómputo

### 5.8 Conclusiones

Este despliegue ya fue el definitivo y se realizó utilizando toda la experiencia adquirida en el primero.

Una vez realizadas las pruebas pertinentes y terminado el parón de verano de la Universidad, comenzó a dar servicio tanto a usuarios internos del Instituto como a usuarios externos de diferentes proyectos europeos.

La tarea de administración de la infraestructura fue simplificada todavía más gracias a Neutron, el nuevo servicio de redes que permitía a los usuarios definir sus propias redes y evitaba tener que reconfigurar el hardware de la red de la infraestructura cada vez que se quería dar soporte a un nuevo grupo de usuarios en un proyecto determinado.

Por parte del autor, se consiguió todavía más experiencia con las herramientas mencionadas, así como con otras nuevas que se introdujeron en este despliegue.

# 6 - Conclusiones

---

## 6.1 Cumplimiento de objetivos

El objetivo principal, que era migrar la infraestructura grid existente a una infraestructura de computación en la nube, fue cumplido con creces.

Al cumplir este objetivo, se cumplió el otro objetivo que estábamos buscando, que era simplificar la tarea de administración de la infraestructura. Hoy en día, ambas infraestructuras requieren una atención mínima por parte de los administradores gracias a la migración a esta nueva infraestructura y a las herramientas de monitorización desplegadas, que avisan de cualquier problema a los administradores para aislarlo y solucionarlo lo más rápidamente posible.

Ambas infraestructuras están funcionando sin problemas, dando servicio tanto a investigadores del propio instituto, de la universidad, y a varios proyectos con financiación europea que se han conseguido gracias a ambas infraestructuras.

## 6.2 Ampliaciones y mejoras para el futuro (CESAR)

Todos los problemas de las despliegues fueron tenidos en cuenta a la hora de diseñar el proyecto **CESAR**.

El proyecto CESAR – Centro de Supercomputación de Aragón - necesitaría otra memoria para él. El BIFI, recibió para el área de Grid & Cloud Computing en el que me encuentro trabajando, 600.000€ de fondos **FEDER** – *Fondo Europeo de Desarrollo Regional* – para la compra de una nueva infraestructura de hardware. Así pues, mientras se realizaban ambos despliegues, se tuvieron en cuenta las oportunidades de mejora de ambos, así como los puntos únicos de fallo, de cara a solucionarlos todos con esta nueva infraestructura, que esta vez sí, estaría pensada específicamente para dar servicio a una infraestructura de computación distribuida en la nube privada basada en OpenStack.

Sus características más destacadas son las siguientes:

- **Mayor potencia de computo:** Procesadores más rápidos, con un mayor número de cores por procesador y más memoria RAM, lo que permite a su vez crear un mayor número de máquinas virtuales.
- **Mayor número de servidores:** Con el fin de tener recursos suficientes para duplicar todos los servicios, ahora hay 8 servidores dedicados única y exclusivamente a albergar los servicios básicos.
- **Servidores dedicados única y exclusivamente al almacenamiento:** Esta nueva infraestructura dispone de 4 servidores dedicados únicamente a dar servicio a Ceph, con el objetivo de que todo el almacenamiento, imágenes, dispositivos de bloques e incluso máquinas virtuales, estén en dicho almacenamiento de modo que todos los datos, incluidas también las máquinas virtuales, estén a salvo de posibles fallos. A destacar de estos servidores de almacenamiento, es que para garantizar el buen funcionamiento de Ceph, disponen de conectividad Infiniband

## 6 - Conclusiones

---

con un gran ancho de banda y una mínima latencia para que las réplicas de los datos almacenados en Ceph se hagan de la manera más rápida posible.

- **Todos los nodos de la infraestructura disponen de su propio RAID 1:** Este era el fallo más típico de los nodos de cómputo de la infraestructura, por encima de otros fallos típicos como fallos en las fuentes de alimentación, o fallos en la placa. Esto suponía un problema especialmente grave, ya que las instancias que estuvieran funcionando en ese momento se perdían, motivo por el cual se decidió añadir un disco duro a cada nodo de cómputo.

Está pendiente de resolución otra petición de fondos FEDER con el objetivo de adquirir un grupo electrógeno para evitar una caída de la infraestructura en el caso de un corte de luz prolongado.

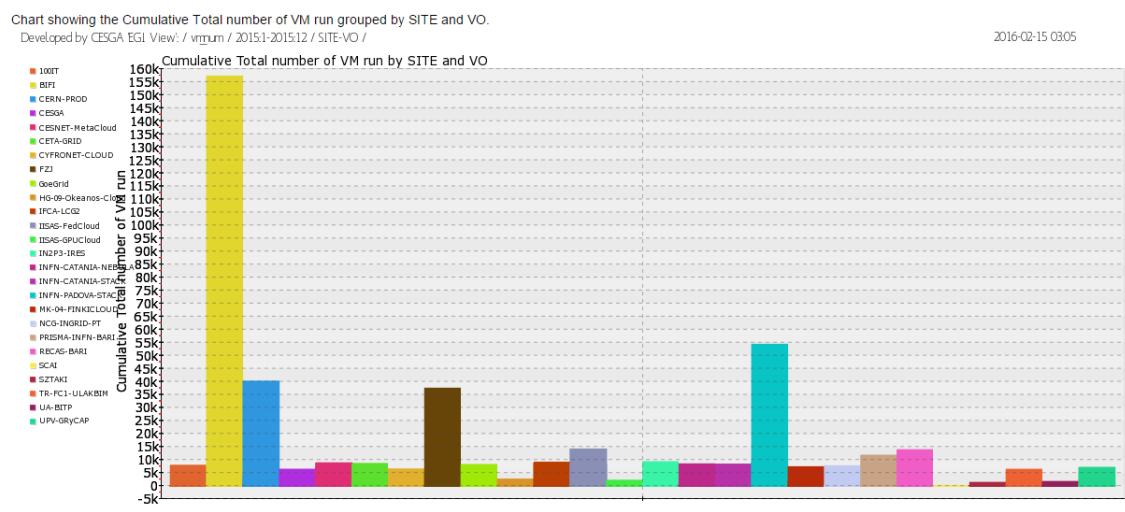
### 6.3 Seguimiento de la infraestructura

Como he mencionado antes, ambas infraestructuras se encuentran hoy en día funcionando y dando servicio a cientos de usuarios, bien sean usuarios del propio instituto, usuarios de la Universidad de Zaragoza, o bien usuarios de distintos proyectos europeos que suponen una importante cantidad de financiación para el BIFI, entre ellos:

- **Cloudflow:** 35.000€
- **CloudSME:** 265.000€
- **EGI-Engage:** 100.000€
- **Fortissimo:** 125.000€

Aparte de la financiación obtenida por dichos proyectos, cabe destacar el amplio uso que se ha hecho de ambas infraestructuras. En concreto, dentro del proyecto EGI-Engage, se han creado durante el año 2015, más de 155.000 máquinas virtuales lo que nos convierte en líderes durante ese año en número de máquinas virtuales creadas, tal y como refleja la **figura 6**

## 6 - Conclusiones



*Figura 6 - Datos de uso de la infraestructura (EGI Accounting Portal)*

En esta figura podemos ver representados a los distintos participantes en el proyecto de EGI-Engage. El BIFI representado por la barra amarilla aparece claramente destacado sobre los demás, en lo referido a máquinas virtuales creadas en la infraestructura.

### 6.4 Incidencias

El comienzo de este PFC fue bastante duro por la temática del mismo y la falta de formación específica en este tema durante la carrera. Si bien de la parte de grid la documentación era más extensa y madura, la de la parte de computación en la nube era más escasa. Pero conforme fue pasando el tiempo y crecía mi experiencia con la plataforma, mi productividad comenzó a aumentar de forma notable y eficiente.

Las mayores incidencias tuvieron que ver con temas ajenos a este PFC, como por ejemplo, el hecho de que en Agosto la Universidad haga un apagón para el mantenimiento de las líneas eléctricas, lo cual afecta negativamente a nuestras infraestructuras, o fallos de hardware en las máquinas de EPSH, lo cual ha hecho que haya tenido que desplazarme hasta allí físicamente en más de una ocasión con la consiguiente pérdida de tiempo.

### 6.5 Valoraciones

La ejecución completa de este PFC se valora de forma muy positiva, por todos los hechos que le han rodeado.

En primer lugar, tener la oportunidad de trabajar con tecnologías punteras y formarme en un tema que está cada día más y más en auge. Asimismo, por la posibilidad de interaccionar con expertos en la materia a nivel internacional en varios congresos a los que he asistido.

Todo esto, puso a prueba mi capacidad de aprendizaje y los conocimientos adquiridos en la carrera.

Este PFC a su vez sirvió como inicio laboral en el BIFI, donde llevo ya 3 años trabajando como responsable técnico de las infraestructuras de computación en la nube del Instituto.