



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

**“ANÁLISIS COMPARATIVO DE TECNOLOGÍAS DE  
INFRAESTRUCTURA COMO SERVICIO EN CLOUD  
COMPUTING Y SU APLICACIÓN DE UN MODELO PARA LA  
EIS”**

**TESIS DE GRADO**

**PREVIA LA OBTENCIÓN DEL TÍTULO DE:**

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**PRESENTADO POR:**

**ROBERTO CARLOS SALAZAR GUALOTO**

**RIOBAMBA – ECUADOR**

**2013**

## **AGRADECIMIENTO**

*A mi Padre Dios piloto principal compañero fiel de los mejores caminos de victorias, derrotas, luchas constantes de la vida.*

*A mis Madres Zenata (+) y Lolita quienes desde mis pequeños pasos son las mejores copilotos de mis caminos y sueños, han guiado celestial y terrenalmente todas mis batallas rumbo a lograr las metas propuestas.*

*Mi eterno agradecimiento al Ing. Danilo Pástor Ing. Gloria Arcos e Ing. Raúl Rosero por impartir sus conocimientos, ideas y colaboración desinteresada para el desarrollo de mi Tesis*

*Roberto Carlos Salazar Gualoto*

## **DEDICATORIA**

*Dedico el presente trabajo de grado a mi Padre Dios  
compañero incansable en cada cruzada de la vida.*

*A mis Madres Zenata (+) y Lolita por nunca declinar en la  
dura labor de formarme como persona y profesional, por ser la guía  
tesonera y firme en cada momento de mi vida.*

*Roberto Carlos Salazar Gualoto*

## FIRMAS RESPONSABILIDAD Y NOTA

NOMBRE	FIRMA	FECHA
Ing. Iván Menes Camejo <b>DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA</b>	-----	-----
Ing. Raúl Rosero <b>DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS</b>	-----	-----
Ing. Danilo Pástor <b>DIRECTOR DE TESIS</b>	-----	-----
Ing. Raúl Rosero <b>MIEMBRO DEL TRIBUNAL</b>	-----	-----
Tlgo. Carlos Rodríguez <b>DIRECTOR DEL CENTRO DE DOCUMENTACIÓN</b>	-----	-----
<b>NOTA DE LA TESIS</b>	-----	

## **RESPONSABILIDAD DEL AUTOR**

Yo **Roberto Carlos Salazar Gualoto**, soy el responsable de las ideas, doctrinas y resultados expuestos en esta Tesis de Grado, y el patrimonio intelectual de la misma pertenece a la “**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**”

---

Roberto Carlos Salazar Gualoto

## ÍNDICE DE ABREVIATURAS

<b>API</b>	Application Programming Interface (Interfaz de Programación de Aplicaciones)
<b>AWS</b>	Amazon Web Services
<b>BPaaS</b>	Business Process as a Service (Procesos de Negocios como Servicio)
<b>CPU</b>	Central Processing Unit, (Unidad Central de Procesamiento)
<b>CRM</b>	Customer Relationship Managemen (Administración Basada en Relación)
<b>DHCP</b>	Dynamic Host Configuration Protocol (Protocolo de Configuración Dinámica de Host)
<b>EBS</b>	Elastic Block Store (Bloque de Almacenamiento Elástico)
<b>EIS</b>	Escuela de Ingeniería en Sistemas
<b>ERP</b>	Enterprise Resource Planning (Planificación de Recursos Empresariales)
<b>ESPOCH</b>	Escuela Superior Politécnica de Chimborazo
<b>GPL</b>	General Public License (Licencia Pública General)
<b>HTTP</b>	Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto)
<b>IaaS</b>	Infrastructure as a Service (Infraestructura como Servicio)
<b>IDS</b>	Intrusion Detection System (Sistema de Detección de Intrusos)
<b>IPS</b>	Intrusion Prevention System (Sistema de Prevención de Intrusos)
<b>iSCSI</b>	internet Small Computer System Interface
<b>LGPL</b>	Lesser General Public License (Licencia Menor Pública General)
<b>NIST</b>	National Institute of Standards and Technology (Instituto Nacional de Estándares y Tecnología)

<b>PaaS</b>	Platform as a Service (Plataforma como Servicio)
<b>REST</b>	REpresentational State Transfer (Transferencia de Estado Representacional)
<b>SaaS</b>	Software as a Service (Software como Servicio)
<b>SO</b>	Sistema Operativo
<b>SOAP</b>	Simple Object Access Protocol (Protocolo Simple de Acceso de Objetos)
<b>TI</b>	Tecnología Informática
<b>TIC</b>	Tecnología de la Información y Comunicación
<b>UIT</b>	Unión Internacional de Telecomunicaciones
<b>VM</b>	Virtual Machine (Máquina Virtual)
<b>VMM</b>	Virtual Machine Manager (Administrador de Máquinas Virtuales)
<b>VPN</b>	Virtual Private Network (Red Privada Virtual)

# ÍNDICE GENERAL

**AGRADECIMIENTO**

**DEDICATORIA**

**FIRMAS RESPONSABILIDAD Y NOTA**

**RESPONSABILIDAD DEL AUTOR**

**ÍNDICE DE ABREVIATURAS**

**ÍNDICE GENERAL**

**ÍNDICE DE FIGURAS**

**ÍNDICE DE TABLAS**

**INTRODUCCIÓN**

**CAPÍTULO I**

Marco Referencial .....	25
1.1. Antecedentes .....	25
1.2. Justificación del Proyecto de Tesis .....	28
1.2.1. Justificación Teórica .....	28
1.2.2. Justificación Práctica .....	30
1.3. Objetivos .....	31
1.3.1. Objetivo General .....	31
1.3.2. Objetivos Específicos .....	32
1.4. Hipótesis .....	32

**CAPÍTULO II**

Marco Teórico Referencial .....	33
2.1. Introducción .....	33
2.2. Cloud Computing .....	33
2.2.1. Conceptualización .....	34

2.2.2.	Evolución de Cloud Computing .....	36
2.2.3.	Clasificación de soluciones Cloud Computing .....	38
2.2.3.1.	Modelos de Servicio .....	39
2.2.3.1.1.	Infrastructure as a Service (IaaS).....	39
2.2.3.1.2.	Platform as a Service (PaaS) .....	39
2.2.3.1.3.	Software as a Service (SaaS).....	40
2.2.3.1.4.	Business Process as a Service (BPaaS).....	41
2.2.3.2.	Tipos de Cloud Computing .....	41
2.2.3.2.1.	Cloud Público (Externo).....	41
2.2.3.2.2.	Cloud Privado (Interno) .....	43
2.2.3.2.3.	Cloud de Comunidad .....	44
2.2.3.2.4.	Cloud Híbrido.....	45
2.3.	Hipervisor .....	46
2.3.1.	Tipo de Hipervisores.....	47
2.3.1.1.	Hipervisores de Tipo 1 o Bare-Metal.....	47
2.3.1.1.1.	XEN .....	48
2.3.1.1.2.	KVM .....	48
2.3.1.2.	Hipervisores de Tipo 2 o Hosted .....	49
 CAPÍTULO III		
Análisis Comparativo de las Tecnologías de Infraestructura como Servicio Open Source para Nubes Privadas en Cloud Computing.....		
		50
3.1.	Introducción .....	50
3.2.	Determinación de las Tecnologías a comparar .....	50
3.3.	Análisis de las Tecnologías de Infraestructura como Servicio seleccionadas .....	54
3.3.1.	Eucalyptus .....	54
3.3.1.1.	Generalidades.....	54

3.3.1.2.	Arquitectura .....	57
3.3.1.2.1.	Interna .....	57
3.3.1.2.2.	Infraestructura .....	57
3.3.2.	Nimbus .....	60
3.3.2.1.	Generalidades.....	60
3.3.2.2.	Arquitectura .....	61
3.3.2.2.1.	Interna .....	61
3.3.2.2.2.	Infraestructura .....	61
3.3.3.	OpenNebula .....	64
3.3.3.1.	Generalidades.....	64
3.3.3.2.	Arquitectura .....	65
3.3.3.2.1.	Interna .....	65
3.3.3.2.2.	Infraestructura .....	67
3.4.	Determinación de parámetros de comparación .....	68
3.4.1.	Configuración .....	69
3.4.2.	Flexibilidad.....	69
3.4.3.	Gestión de Usuarios.....	69
3.4.4.	Gestión de Nodos .....	70
3.4.5.	Gestión de Máquina Virtuales.....	70
3.4.6.	Indicadores parámetro Configuración .....	71
3.4.7.	Indicadores parámetro Flexibilidad.....	73
3.4.8.	Indicadores parámetro Gestión de Usuarios .....	74
3.4.9.	Indicadores parámetro Gestión de Nodos .....	75
3.4.10.	Indicadores parámetro Gestión de Máquinas Virtuales .....	76
3.5.	Descripción de los Módulos de Prueba .....	77
3.5.1.	Módulo 1 .....	78

3.5.2.	Módulo 2 .....	78
3.5.3.	Módulo 3 .....	78
3.5.4.	Módulo 4 .....	78
3.5.5.	Módulo 5 .....	79
3.6.	Desarrollo de los Módulos de Prueba .....	79
3.6.1.	Desarrollo de Módulos de Prueba en la IaaS Eucalyptus .....	79
3.6.1.1.	Módulo 1 .....	79
3.6.1.2.	Módulo 2 .....	81
3.6.1.3.	Módulo 3 .....	82
3.6.1.4.	Módulo 4 .....	83
3.6.1.5.	Módulo 5 .....	84
3.6.2.	Desarrollo de Módulos de Prueba en la IaaS Nimbus .....	85
3.6.2.1.	Módulo 1 .....	85
3.6.2.2.	Módulo 2 .....	87
3.6.2.3.	Módulo 3 .....	88
3.6.2.4.	Módulo 4 .....	89
3.6.2.5.	Módulo 5 .....	90
3.6.3.	Desarrollo de Módulos de Prueba en la IaaS OpenNebula .....	91
3.6.3.1.	Módulo 1 .....	91
3.6.3.2.	Módulo 2 .....	93
3.6.3.3.	Módulo 3 .....	94
3.6.3.4.	Módulo 4 .....	95
3.6.3.5.	Módulo 5 .....	96
3.7.	Análisis Comparativo .....	97
3.7.1.	Parámetro Configuración .....	99
3.7.1.1.	Determinación de Variables .....	99

3.7.1.2.	Valoraciones .....	99
3.7.1.3.	Interpretación.....	100
3.7.1.4.	Calificación .....	101
3.7.2.	Parámetro Flexibilidad .....	102
3.7.2.1.	Determinación de Variables .....	102
3.7.2.2.	Valoraciones .....	102
3.7.2.3.	Interpretación.....	103
3.7.2.4.	Calificación .....	104
3.7.3.	Parámetro Gestión de Usuarios .....	104
3.7.3.1.	Determinación de Variables .....	104
3.7.3.2.	Valoraciones .....	105
3.7.3.3.	Interpretación.....	105
3.7.3.4.	Calificación .....	106
3.7.4.	Parámetro Gestión de Nodos .....	107
3.7.4.1.	Determinación de Variables .....	107
3.7.4.2.	Valoraciones .....	107
3.7.4.3.	Interpretación.....	107
3.7.4.4.	Calificación .....	108
3.7.5.	Parámetro Gestión de Máquinas Virtuales .....	108
3.7.5.1.	Determinación de Variables .....	108
3.7.5.2.	Valoraciones .....	109
3.7.5.3.	Interpretación.....	109
3.7.5.4.	Calificación .....	110
3.7.6.	Puntajes Alcanzados.....	111
3.7.7.	Interpretación.....	114
3.7.8.	Resultado del Análisis.....	114

3.7.9.	Conclusión .....	116
3.7.10.	Comprobación de Hipótesis .....	116

## CAPÍTULO IV

Implementación de Prototipo de Infraestructura como Servicio “EIS” .....	122	
4.1.	Introducción .....	122
4.2.	Instalación y Administración Eucalyptus .....	122
4.2.1.	Requerimientos Hardware y Software .....	122
4.3.	Instalación de IaaS .....	123
4.3.1.	Instalación en el Front End .....	123
4.3.1.1.	Instalación de Dependencias .....	123
4.3.1.1.1.	Configuración del Repositorio YUM para Eucalyptus Front End .....	123
4.3.1.1.2.	Instalación de Eucalyptus .....	124
4.3.1.1.3.	Configuración de Eucalyptus .....	124
4.3.1.1.3.1.	Registro de Walrus .....	124
4.3.1.1.3.2.	Registro del Cluster .....	125
4.3.1.1.3.3.	Registro Storage Controller .....	125
4.3.1.1.4.	Instalación de Herramientas de Usuario .....	125
4.3.1.1.4.1.	Configuración de Repositorio .....	125
4.3.1.1.4.2.	Instalación de herramientas para el Usuario .....	126
4.3.1.1.4.3.	Configuración .....	126
4.3.2.	Instalación del Nodo XEN .....	128
4.3.2.1.	Instalación de Dependencias .....	128
4.3.2.1.1.	Instalación de Hypervisor XEN .....	128
4.3.2.1.2.	Configurar XEN .....	130
4.3.2.1.3.	Configuración del Repositorio YUM para Eucalyptus Nodo .....	130
4.3.3.	Instalación de Eucalyptus en el Nodo .....	130

4.3.3.1.	Registro de Node Controller.....	132
4.4.	Administración de Eucalyptus .....	132
4.4.1.	Registrar el Kernel .....	132
4.4.2.	Registro Ramdisk.....	134
4.4.3.	Gestión de Imágenes.....	135
4.4.3.1.	Registro de Imagen Centos.....	135
4.4.3.2.	Eliminación de Imagen .....	137
4.4.3.3.	Descargar Imagen desde Repositorio.....	137
4.4.3.4.	Mostrar Información de las Imágenes Registradas.....	137
4.4.4.	Gestión de Máquinas Virtuales .....	138
4.4.4.1.	Creación de Keypairs .....	138
4.4.4.2.	Creación Máquina Virtual con SO Centos .....	138
4.4.4.3.	Iniciar Máquina Virtual Centos.....	139
4.4.4.4.	Reinicio de Máquinas Virtuales .....	140
4.4.4.5.	Mostrar Máquinas Virtuales en Ejecución .....	140
4.4.4.6.	Monitorear Máquina Virtual .....	140
4.4.4.7.	Apagar Máquinas Virtuales .....	141
4.4.4.8.	Creación de Imagen Windows.....	141
4.4.4.8.1.	Instalación Complemento de Integración Eucalyptus Windows.....	143
4.4.4.8.2.	Registro de Imagen Windows.....	145
4.4.4.8.3.	Iniciar Máquina Virtual Windows.....	147

**CONCLUSIONES**

**RECOMENDACIONES**

**RESUMEN**

**SUMMARY**

**GLOSARIO**

**ANEXOS**

**BIBLIOGRAFÍA**

## ÍNDICE DE FIGURAS

Figura I - 1. Modelos de Servicio de Cloud Computing .....	26
Figura I - 2. Modelo de Infraestructura como Servicio para la EIS .....	31
Figura II - 3. Evolución del Cloud Computing .....	37
Figura II - 4. Cubo Cloud Computing.....	38
Figura II - 5. Arquitectura de Trabajo de un Hipervisor.....	47
Figura II - 6. Tipo de Hipervisores.....	49
Figura III - 7. Resultados Encuesta Aceptación IaaS.....	53
Figura III - 8. Logo de Eucalyptus.....	54
Figura III - 9. Arquitectura IaaS Eucalyptus .....	58
Figura III - 10. Logo Nimbus.....	60
Figura III - 11. Arquitectura IaaS Nimbus .....	61
Figura III - 12. Logo OpenNebula.....	64
Figura III - 13. Capas OpenNebula .....	64
Figura III - 14. Arquitectura IaaS OpenNebula.....	65
Figura III - 15. Componentes OpenNebula.....	67
Figura III - 16. Comparación Porcentajes Parámetro Configuración.....	101
Figura III - 17. Comparación Porcentajes Parámetro Flexibilidad.....	104
Figura III - 18. Comparación Porcentajes Parámetro Gestión de Usuarios.....	106
Figura III - 19. Comparación Porcentajes Parámetro Gestión de Nodos .....	108
Figura III - 20. Comparación Porcentajes Parámetro Gestión de Máquinas Virtuales.....	111
Figura III - 21. Comparación General de Porcentajes.....	113
Figura III - 22. Comparación Final de Porcentajes.....	114
Figura III - 23. Comparación Final Porcentajes Sin y Con IaaS .....	121
Figura IV – 24. Proceso de Iniciación de Servicio de Eucalyptus.....	124

Figura IV – 25. Descarga de Credenciales .....	126
Figura IV – 26. Inicio Servicio Node Controller.....	131
Figura IV – 27. Comprobación de iteración de libvirt y el nodo de computación.....	131
Figura IV – 28. Descarga Imagen Certificada de Centos .....	132
Figura IV – 29. Registro de kernel de la Imagen Certificada de Centos .....	134
Figura IV – 30. Registro de ramdisk de la Imagen Certificada de Centos.....	135
Figura IV – 31. Registro de la Imagen Certificada de Centos .....	136
Figura IV – 32. Listado de Imágenes Registradas .....	137
Figura IV – 33. Información de Máquinas Virtuales Creada.....	138
Figura IV – 34. Tipos de Máquinas Virtuales.....	139
Figura IV – 35. Inicio de Máquina Virtual .....	139
Figura IV – 36. Listado de Máquinas Virtuales en ejecución.....	140
Figura IV – 37. Monitoreo Máquina Virtual .....	140
Figura IV – 38. Apagar Máquina Virtual .....	141
Figura IV – 39. Preparar Instalación Eucalyptus Integration Windows.....	143
Figura IV – 40. Inicio Instalación Eucalyptus Integration Windows .....	144
Figura IV – 41. Ubicación Archivo Eucalyptus Integration Windows .....	144
Figura IV – 42. Hypervisor Eucalyptus Integration Windows .....	144
Figura IV – 43. Instalación Correcta Eucalyptus Integration Windows .....	145
Figura IV – 44. Registro de Imagen Windows .....	146
Figura IV – 45. Registro bucket de Imagen Windows .....	146
Figura IV – 46. Información Imagen Windows .....	147
Figura IV – 47. Iniciar VM Windows .....	148
Figura IV – 48. Conexión remoto VM Window .....	148
Figura IV – 49. Autenticación VM Window .....	148
Figura IV – 50. Verificación Certificados de Seguridad VM Window .....	149

Figura IV – 51. Escritorio Remoto VM Window ..... 149

## ÍNDICE DE TABLAS

Tabla III. I Porcentaje de pesos de los parámetros de evaluación.....	71
Tabla III. II Indicador Grado de Complejidad.....	71
Tabla III. III Indicador Nivel de Conocimiento .....	72
Tabla III. IV Indicador Cantidad de Tiempo .....	72
Tabla III. IV Indicador Cantidad de Recurso Humano.....	73
Tabla III. V Indicador Cantidad Soporte Hypervisor .....	73
Tabla III. VI Indicador Grado de Uso de DHCP.....	73
Tabla III. VII Indicador Nivel de Despliegue.....	74
Tabla III. VIII Indicador Cantidad de Entorno de Programación.....	74
Tabla III. IX Indicador Creación, Modificación, Eliminación de Usuarios .....	74
Tabla III. X Indicador Monitoreo de Usuarios.....	75
Tabla III. XI Indicador Protocolo de Seguridad .....	75
Tabla III. XII Indicador Creación, Configuración y Eliminación de Nodos.....	75
Tabla III. XIII Indicador Monitoreo de Nodos .....	76
Tabla III. XIV Indicador Creación, Configuración y Eliminación de Máquinas Virtuales.....	76
Tabla III. XV Indicador Iniciar, Reiniciar, Mostrar y Apagar Máquinas Virtuales.....	76
Tabla III. XVI Indicador Monitoreo de Máquinas Virtuales .....	77
Tabla III. XVII Indicador Variedad de Sistemas Operativos .....	77
Tabla III. XVIII Número de pasos a ejecutar Eucalyptus .....	80
Tabla III. XIX Índices de Conocimiento Eucalyptus.....	80
Tabla III. XX Cantidad de Tiempo Eucalyptus .....	81
Tabla III. XXI Personal Técnico necesario Eucalyptus.....	81
Tabla III. XXII Soporte Hypervisores Eucalyptus.....	81
Tabla III. XXIII Entornos de Programación Eucalyptus.....	82

Tabla III. XXIV Número de Comandos para Creación, Modificación, Eliminación de Usuarios Eucalyptus .....	82
Tabla III. XXV Número de Comandos Monitoreo Usuarios Eucalyptus.....	83
Tabla III. XXVI Número de Comandos para Creación, Configuración, Eliminación de Nodos Eucalyptus .....	83
Tabla III. XXVII Número de Comandos Monitoreo de Nodos Eucalyptus .....	84
Tabla III. XXVIII Número de Comandos para Creación, Configuración, Eliminación de Máquinas Virtuales Eucalyptus.....	84
Tabla III. XXIX Número de Comandos para Iniciar, Reiniciar, Mostrar y Apagar Máquinas Virtuales Eucalyptus.....	84
Tabla III. XXX Número de Comandos para Monitoreo de Máquinas Virtuales Eucalyptus .....	85
Tabla III. XXXI Número de pasos a ejecutar Nimbus .....	86
Tabla III. XXXII Índices de Conocimiento Nimbus.....	86
Tabla III. XXXIII Cantidad de Tiempo Nimbus.....	87
Tabla III. XXXIV Personal Técnico necesario Nimbus.....	87
Tabla III. XXXV Soporte Hypervisores Nimbus.....	87
Tabla III. XXXVI Entornos de Programación Eucalyptus .....	88
Tabla III. XXXVII Número de Comandos para Creación, Modificación, Eliminación de Usuarios Nimbus .....	88
Tabla III. XXXVIII Número de Comandos Monitoreo Usuarios Nimbus.....	89
Tabla III. XXXIX Número de Comandos para Creación, Configuración, Eliminación de Nodos Nimbus .....	89
Tabla III. XXXX Número de Comandos Monitoreo de Nodos Nimbus .....	90
Tabla III. XXXXI Número de Comandos para Creación, Configuración, Eliminación de Máquinas Virtuales Nimbus.....	90
Tabla III. XXXXII Número de Comandos para Iniciar, Reiniciar Mostrar y Apagar Máquinas Virtuales Nimbus.....	90
Tabla III. XXXXIII Número de Comandos para Monitoreo de Máquinas Virtuales Nimbus .....	91
Tabla III. XXXXIV Número de pasos a ejecutar OpenNebula.....	92

Tabla III. XXXXV Índices de Conocimiento OpenNebula.....	92
Tabla III. XXXXVI Cantidad de Tiempo OpenNebula.....	92
Tabla III. XXXXVII Personal Técnico necesario OpenNebula .....	93
Tabla III. XXXXVIII Soporte Hypervisores Eucalyptus .....	93
Tabla III. XXXXIX Entornos de Programación OpenNebula .....	94
Tabla III. XXXXX Número de Comandos para Creación, Modificación, Eliminación de Usuarios OpenNebula .....	94
Tabla III. XXXXXI Número de Comandos Monitoreo OpenNebula.....	95
Tabla III. XXXXXII Número de Comandos para Creación, Configuración, Eliminación de Nodos OpenNebula .....	95
Tabla III. XXXXXIII Número de Comandos Monitoreo de Nodos OpenNebula .....	96
Tabla III. XXXXXIV Número de Comandos para Creación, Configuración, Eliminación de Máquinas Virtuales OpenNebula .....	96
Tabla III. XXXXXV Número de Comandos para Iniciar, Reiniciar Mostrar y Apagar Máquinas Virtuales OpenNebula .....	96
Tabla III. XXXXXVI Número de Comandos para Monitoreo de Máquinas Virtuales OpenNebula ....	97
Tabla III. XXXXXVII Escala Cualitativa para los Indicadores .....	98
Tabla III. XXXXXVIII Escala Cuantitativa para los Indicadores.....	98
Tabla III. XXXXXIX Valoración Parámetro Configuración.....	100
Tabla III. XXXXXX Valoración Parámetro Flexibilidad .....	103
Tabla III. XXXXXI Valoración Parámetro Gestión de Usuarios.....	105
Tabla III. XXXXXII Valoración Parámetro Gestión de Nodos.....	107
Tabla III. XXXXXIII Valoración Parámetro Gestión de Máquinas Virtuales.....	109
Tabla III. XXXXXIV Tabla General de Resultados .....	112
Tabla III. XXXXXV Parámetros de Hipótesis.....	117
Tabla III. XXXXXVI Valores Parámetros de Hipótesis.....	118
Tabla III. XXXXXVII Valores y Porcentajes Finales.....	120
Tabla IV. XXXXXVIII Requerimientos Hardware .....	122

Tabla IV. XXXXXXIX Requerimientos Software .....	123
--	-----

## INTRODUCCIÓN

Cloud Computing es un modelo para permitir omnipresente, conveniente y bajo demanda de acceso de red a un conjunto compartido de recursos informáticos configurables por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios que pueden ser rápidamente provisionados y liberados con una mínima gestión esfuerzo o interacción con el proveedor o administrador de servicio.

Tipos de modelos de *cloud computing* están nube pública, nube privada y nube híbrida. En la nube pública los recursos se proporcionan a través de Internet a todos los clientes. En la nube privada los recursos se proporcionan a través de la intranet de una organización. Nube híbrida es una disposición según requerimiento puede proporcionar recursos dentro de una organización o público.

Tipos de servicios de *cloud computing* incluyen IaaS, PaaS y SaaS. En IaaS Infraestructura como un servicio las máquinas virtuales, almacenamiento bruto (bloque), firewalls, balanceadores de carga, y las redes se proporcionan como servicios basados en la nube a los clientes. PaaS Plataforma como Servicio es una forma de ofrecer plataformas de los sistemas operativos, plataformas de desarrollo de aplicaciones. SaaS Software como Servicio permite al proveedor de servicio dar software como Enterprise Resource Planning (ERP), gestión de relaciones con clientes (CRM) a los usuarios.

El presente trabajo de grado se encuentra estructurado en cuatro capítulos más un apartado final de conclusiones recomendaciones glosario resumen summary anexos y bibliografía. El primer capítulo hace referencia a la descripción general de la información del presente proyecto de tesis, en el segundo capítulo está el fundamento teórico necesario para el análisis comparativo, el tercer capítulo es el más importante contiene la información de las tecnologías de Infraestructura como Servicio Open Source para Nubes Privadas en Cloud Computing a analizar, los parámetros de evaluación, análisis comparativo y comprobación de la hipótesis. El Cuarto capítulo contiene la implementación de un Prototipo con Tecnología de Infraestructura como Servicio Open Source para Nubes Privadas en Cloud Computing.

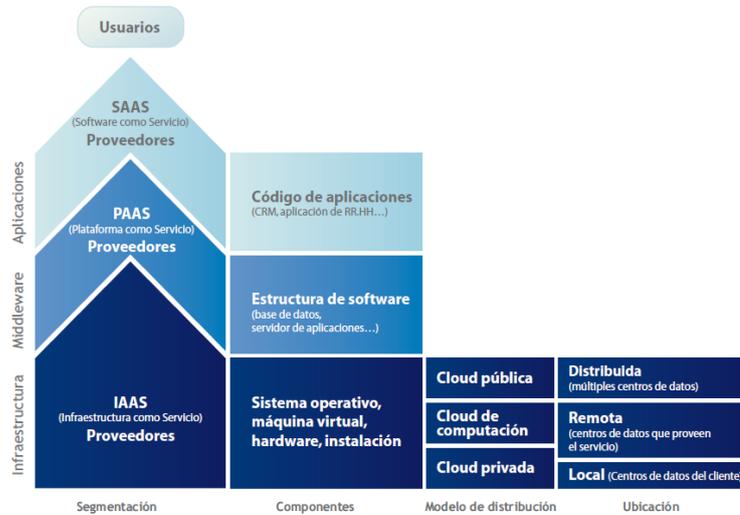
## CAPÍTULO I

### Marco Referencial

#### 1.1. Antecedentes

En la actualidad el término “*Cloud Computing*” ha impactado al mundo de la tecnología. Wikipedia define *Cloud Computing* como “un paradigma de computación en el cual recursos dinámicamente escalables y frecuentemente virtualizados se proveen como servicio sobre la red de Internet”. (1)

En este caso, la **Infraestructura como Servicio (IaaS)** es el eje central para que la “nube” pueda funcionar correctamente. La IaaS es donde se inicia la construcción de toda la infraestructura del Cloud Computing, y que da paso a la **Plataforma como Servicio (PaaS)** y al **Software como Servicio (SaaS)**. (5)



**Figura I - 1. Modelos de Servicio de Cloud Computing**

La **Infraestructura como Servicio (IaaS)**, contempla la entrega de servicios, también denominados servicios computacionales fundamentales, entre los cuales se encuentran: almacenamiento, procesamiento y memoria. Dicha infraestructura es desplegada bajo demanda, permitiendo a los usuarios el despliegue de aplicaciones sobre un sistema operativo principal. En este modelo de servicio el usuario final no administra ni controla la infraestructura base *cloud computing*, pero puede controlar dispositivos de almacenamiento, sistemas operativos, aplicaciones desplegadas y opcionalmente controlar componentes de red, tales como un firewall o un enrutador.

En el mercado de la Infraestructura como Servicio (IaaS) existen una gama variada de tecnologías de las cuales nuestro estudio se interesa por las de código abierto (Open Source) para nubes privadas como son:

- ✓ Eucalyptus, infraestructura open source para la implementación de computación en la nube privada en clusters de ordenadores. Implementa nubes de tipo privado

e híbrido, de estilo IaaS (Infrastructure as a Service), proporciona una interfaz única que permite al usuario acceso a recursos de infraestructura (máquinas, red y almacenamiento).

- ✓ OpenNebula, herramienta de gestión completamente de código abierto para infraestructura como servicio (IaaS) de cloud computing. OpenNebula se utiliza principalmente como una herramienta de virtualización para gestionar su infraestructura virtual en el centro de datos o grupo, que se suele denominar como **nube privada**.
- ✓ Nimbus, conjunto de herramientas de código abierto enfocado en proveer la Infraestructura como Servicio (IaaS) para nubes privadas.
- ✓ OpenStack (Nova), proyecto de Infraestructura como Servicio (IaaS) cloud computing de código abierto diseñado para proporcionar y administrar grandes redes de máquinas virtuales, la creación de una plataforma de Cloud Computing redundante y escalable le da el software, los paneles de control, y las API necesarias para orquestar una nube privada, incluyendo las instancias en ejecución, gestión de redes, y controlar el acceso a través de los usuarios.
- ✓ Red Hat CloudForms, permite crear y administrar la Infraestructura como Servicio (IaaS) en nubes privadas y híbridas mediante la incorporación de la gestión integral del ciclo de vida tanto de aplicaciones y la capacidad de crear nubes integradas de la gama más amplia de los recursos informáticos con la portabilidad única en entornos físicos, virtuales y recursos de cloud computing.

La Infraestructura computacional de la Escuela de Ingeniería en Sistemas es brindada mediante equipos físicos, cuando existe la necesidad de un nuevo sistema operativo para

diferentes prácticas es necesario realizar varias operaciones como particionar el disco, instalar dicho sistema operativo para finalmente que este requerimiento computacional entre en ejecución, razón por la cual es necesario tener un ambiente que integre recursos computacionales para poner a disposición los mismos como un servicio.

Para esto es necesario realizar un análisis comparativo de tecnologías de Infraestructura como Servicio en Cloud Computing de código abierto (Open Source) para Nubes privadas, este análisis nos permitirá determinar las ventajas de proveer infraestructura como servicio y conocer la mejor infraestructura que permita a los estudiantes y docentes de la EIS tener servicios computacionales desplegados bajo demanda.

Los servicios computacionales a entregarse bajo demanda requieren se implemente ambientes de pruebas para que mediante un intranet se tenga máquinas virtuales a disposición de los usuarios. Los sistemas operativos para las máquinas virtuales son Linux Centos 5, Windows 7 SO más utilizados en la escuela y necesarios para diversas prácticas.

## **1.2. Justificación del Proyecto de Tesis**

### **1.2.1. Justificación Teórica**

El Cloud Computing específicamente su Infraestructura como Servicio se ha consolidado en los diferentes sectores económicos, educativos y sociales por las diferentes ventajas que ofrece entre las que se encuentran las siguientes (14)

- ✓ **Focalización.** La infraestructura como servicio permite a quien la contrata liberar tiempo y recursos hasta ese momento destinados a mantener su propia infraestructura y dirigirlos a la base de su negocio.
- ✓ **Ahorro de recursos y de costes.** Al contratar solo lo que se necesita, no hace falta tener por ej. discos más grandes o procesador de mayor capacidad de los que realmente necesitan las aplicaciones, que redundan en un ahorro de costes comparado con el precio de una solución de alta disponibilidad de hardware similar basada en servidores ya sea dedicados o en modalidad de “housing”.
- ✓ **Flexibilidad.** Tanto para disponer de más recursos de procesamiento, almacenamiento o red como para reducirlos. Esto es: disponer justo de lo que se necesita y no tener capacidad (equipos) ociosa.
- ✓ **Reducción del coste de propiedad.** Al pasar de comprar a alquilar y no depender de la obsolescencia del hardware adquirido, se pasa a un modelo más económico a corto, medio y largo plazo.
- ✓ **Green IT.** Como consecuencia de la optimización en el uso de recursos. Una infraestructura Cloud bien diseñada está pensada para que los recursos que no se están utilizando no consuman electricidad ni precisen refrigeración, por ejemplo durante las horas de menor carga.
- ✓ **Seguridad física.** Las soluciones IaaS están basadas en una infraestructura de alta redundancia, tanto en los elementos de electrónica de red como en los elementos servidores. Se trata de una infraestructura diseñada para permitir la continuidad del servicio sin interrupción incluso frente a fallos graves de hardware.

- ✓ **Seguridad lógica.** A través del aislamiento completo entre clientes IaaS y a través de protecciones perimetrales y de segmentos tanto físicos como lógicos (firewalls IDS, IPS, etc).

Por todo esto se ve la necesidad de realizar un análisis comparativo de Tecnologías de Infraestructura como Servicio en Cloud Computing de código abierto (Open Source) para Nubes Privadas y su aplicación de un Modelo Infraestructura como Servicio para la EIS que permitirá:

- ✓ Conocer aspectos fundamentales de los diferentes Tipos de Nubes principalmente las Nubes Privadas.
- ✓ Identificar las Tecnologías de Infraestructuras como Servicio en Cloud Computing de código abierto presentes en el mercado.
- ✓ Determinar las principales funcionalidades que un Modelo de Infraestructura como Servicio (IaaS) presenta.
- ✓ Estudiar los servicios de infraestructura que se puede entregar a los usuarios de la EIS.
- ✓ Verificar la confiabilidad de los servicios computacionales que se van a entregar a los usuarios.

### **1.2.2. Justificación Práctica**

Con el análisis comparativo de Infraestructuras como Servicio en Cloud Computing de código abierto (Open Source) para Nubes Privadas nos proyectamos a escoger una Infraestructura que optimice los recursos computacionales que posee la EIS.

## ESCENARIO PROPUESTO

La implementación de un Modelo de Infraestructura como Servicio para la EIS requiere un ambiente de pruebas, para ello se propone el siguiente escenario.

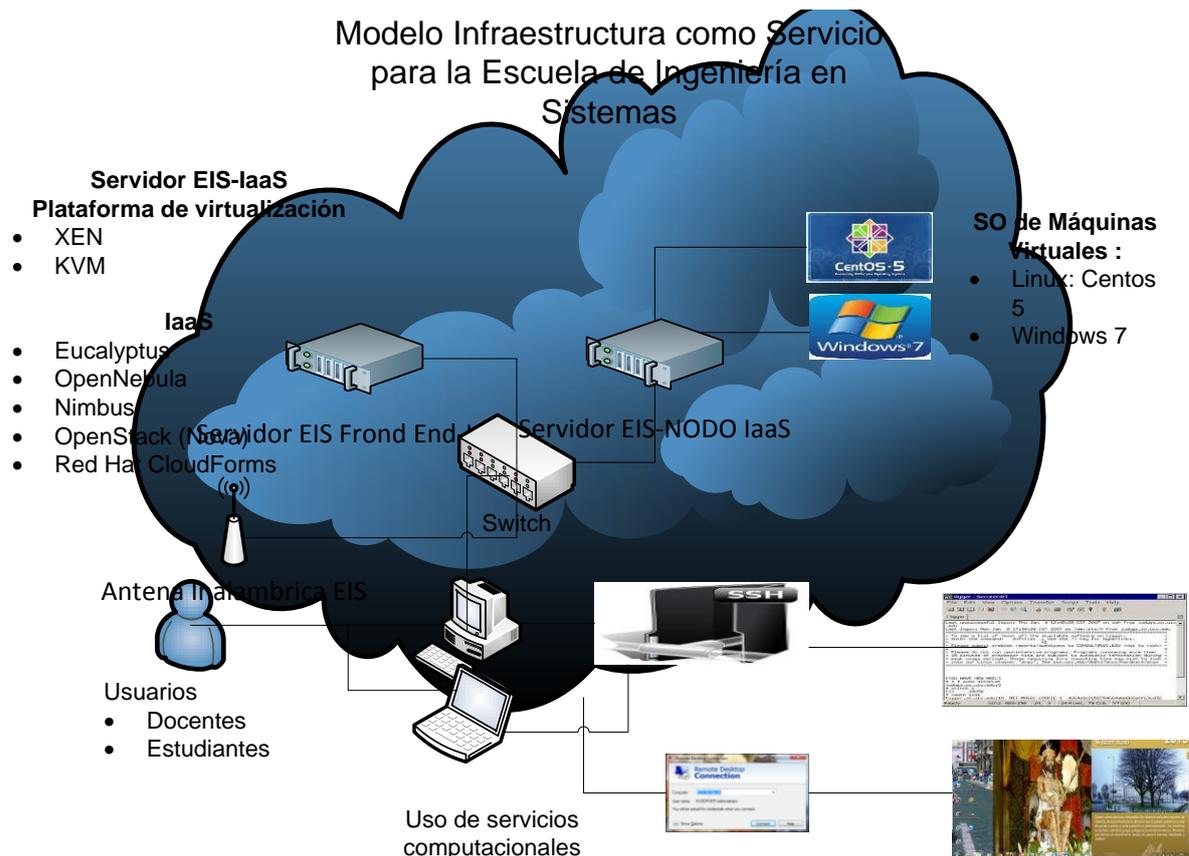


Figura I - 2. Modelo de Infraestructura como Servicio para la EIS

### 1.3. Objetivos

#### 1.3.1. Objetivo General

- ✓ Realizar un análisis comparativo de Tecnologías de Infraestructura como Servicio en Cloud Computing para Nubes Privadas y su aplicación de un modelo de Infraestructura como Servicio para la EIS.

### **1.3.2. Objetivos Específicos**

- ✓ Estudiar las características de las Tecnologías de Infraestructuras como Servicio de código abierto (Open Source) para Cloud Computing.
- ✓ Establecer los parámetros a comparar entre las Infraestructuras como Servicio.
- ✓ Construir ambientes de pruebas para realizar análisis comparativo de cada una de las Tecnologías de Infraestructuras como Servicio en Cloud Computing para Nubes Privadas, para seleccionar la mejor tecnología que se ajuste a las necesidades de la EIS.
- ✓ Implementar un Prototipo de Infraestructura como Servicio para la Escuela de Ingeniería en Sistemas ESPOCH.

### **1.4. Hipótesis**

La Implementación de un modelo de Infraestructura como Servicio Cloud Computing para Nubes Privadas permitirá mejorar la eficiencia de servicios computacionales de la EIS.

## **CAPÍTULO II**

### **Marco Teórico Referencial**

#### **2.1. Introducción**

En este capítulo se abarca los conceptos básicos y términos explicativos fundamentales para el desarrollo del presente tema de tesis. Su finalidad es tener en claro los fundamentos teóricos que se necesita conocer sobre *Cloud Computing* esencialmente en lo que se refiere a Infraestructuras como servicio, lo cual será base para el mejor entendimiento de los términos empleados en los capítulos posteriores.

#### **2.2. Cloud Computing**

En las últimas décadas los procesos de deslocalización e internacionalización de las grandes empresas unidas a la explosión en el uso de tecnologías de información y procesamiento de datos, han hecho que las necesidades de cómputo de las grandes

empresas y organizaciones crezcan a un ritmo superior al que lo hacía la capacidad de cómputo de los ordenadores personales. Por este motivo y para satisfacer las necesidades de los sistemas de computación más exigentes, se ha producido una interesante evolución de las arquitecturas de cómputo, basada fundamentalmente en la ejecución simultánea de procesos en múltiples equipos informáticos.

A lo largo del presente análisis se ha optado por emplear el término en idioma inglés *cloud computing* y no el de computación en nube en idioma español tal y como lo traduce la Unión Internacional de Telecomunicaciones (UIT) y la Unión Europea dado que es el término en idioma inglés el que aparece citado con mayor frecuencia en considerable parte de la literatura especializada y publicaciones consultadas durante la elaboración del mismo.

Para entender mejor el concepto de *cloud computing* y su origen, en este apartado se presenta una breve reseña de la evolución de las arquitecturas de cómputo en las últimas décadas así como, su transformación natural hasta la aparición del paradigma de cloud computing.

### **2.2.1. Conceptualización**

Atendiendo a la definición<sup>1</sup> dada por el NIST (*National Institute of Standards and Technology*), el *cloud computing* es un modelo tecnológico que permite el acceso ubicuo, adaptado y bajo demanda en red a un conjunto compartido de recursos de

---

<sup>1</sup> The NIST Definition of *Cloud Computing*, NIST. ([http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145\\_cloud-definition.pdf](http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf))

computación configurables (por ejemplo: redes, servidores, equipos de almacenamiento, aplicaciones y servicios), que pueden ser rápidamente aprovisionados y liberados con un esfuerzo de gestión reducido o interacción mínima con el proveedor del servicio.

Otra definición complementaria es la aportada por el RAD Lab de la Universidad de Berkeley, donde se explica que el *cloud computing* se refiere tanto a las aplicaciones entregadas como servicio a través de Internet, como el hardware y el software de los centros de datos que proporcionan estos servicios. Los servicios anteriores han sido conocidos durante mucho tiempo como Software as a Service (SaaS), mientras que el hardware y software del centro de datos es a lo que se llama nube<sup>2</sup>.

Con la información sintetizada en este apartado, se entiende que el *cloud computing* representa un cambio importante en cómo pueden las Empresas, Organismos Públicos y Privados procesar la información y gestionar las áreas TIC; apreciándose que con la gestión TIC tradicional las empresas realizan cuantiosas inversiones en recursos, incluyendo *hardware*, *software*, centros de procesamiento de datos, redes, personal, seguridad, etc. Mientras que con los modelos de soluciones en la nube se elimina la necesidad de grandes inversiones y costes fijos, transformando a los proveedores en empresas de servicios que ofrecen de forma flexible e instantánea la capacidad de computación bajo demanda<sup>3</sup>.

---

<sup>2</sup> Above the Clouds: A Berkeley View of *Cloud Computing*l, University of California at Berkeley. (<http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>)

<sup>3</sup> *Cloud Computing: Retos y Oportunidades*l, Fundación Ideas. ([http://www.fundacionideas.es/sites/default/files/pdf/DT-Cloud\\_Computing-Ec.pdf](http://www.fundacionideas.es/sites/default/files/pdf/DT-Cloud_Computing-Ec.pdf))

### 2.2.2. Evolución de Cloud Computing

Debido a las necesidades de cómputo descritas, se ha venido realizando un importante esfuerzo en la investigación de capacidades para la ejecución de procesos en múltiples computadores. Esta tendencia fue impulsada originalmente por la utilización de sistemas abiertos, interoperables y protocolos de comunicación estándar que permitían la comunicación eficiente entre sistemas y tecnologías heterogéneos.

El primer paso de esta evolución fue en gran medida propiciado por los sistemas operativos tipo Unix que permitieron la configuración de *clusters*, es decir, agrupaciones de ordenadores con componentes de hardware comunes que se comportan como un único computador.

Tras varias décadas de investigaciones y desarrollos en estas tecnologías, la irrupción del sistema operativo Linux y sus estándares abiertos permitió implementar *clusters* basados en la arquitectura estándar de los PC, consiguiendo instalaciones de cómputo de alto rendimiento a bajos precios y popularizando esta solución durante la década de los años 1990.

Estos *clusters* sufrieron un proceso de especialización para proporcionar servicios de cómputo y almacenamiento, fundamentalmente en centros de investigación y universidades. Estos centros comenzaron a ofrecer sus servicios a terceros a través de protocolos estándar constituyendo la denominada arquitectura de computación *grid*, orientada al procesamiento en paralelo o al almacenamiento de gran cantidad de información.

Estas arquitecturas fueron acogidas en instituciones investigadoras a principios de la década de los años 2000, pero la complejidad para utilizar la infraestructura, las dificultades para utilizar diferentes *grids*, los problemas de portabilidad entre ellas hicieron que nunca se popularizara, fuera del ámbito de la investigación y académico.

Durante esta misma época comenzaron a popularizarse las tecnologías de virtualización que hacían posible implementar máquinas virtuales que desacoplan el hardware del software y permiten replicar el entorno del usuario sin tener que instalar y configurar todo el software que requiere cada aplicación. Esto tiene ventajas en la distribución y mantenimiento de sistemas de software complejos y permite integrar bajo un mismo entorno un conjunto de sistemas heterogéneos.

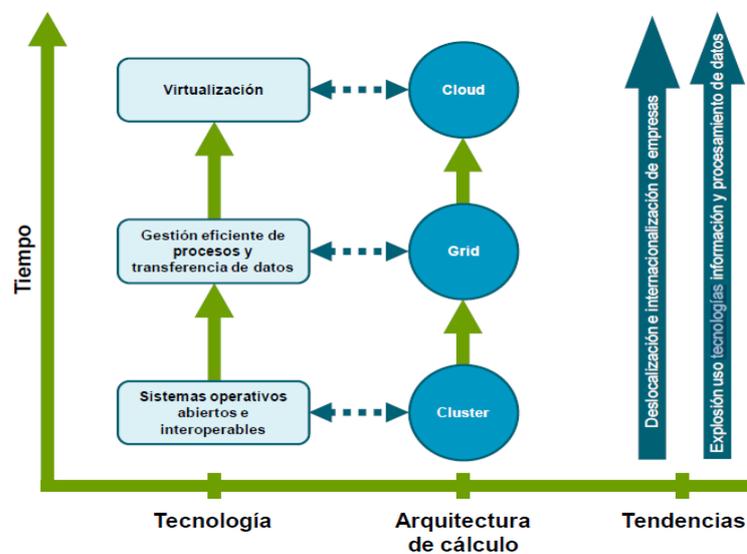


Figura II - 3. Evolución del Cloud Computing

Esta nueva arquitectura permitía distribuir carga de trabajo de forma sencilla, lo cual elimina los problemas que presentaba la arquitectura *grid*, abriendo una nueva puerta a la computación distribuida, llamado *cloud computing*. Este nuevo modelo emerge como

un nuevo paradigma capaz de proporcionar recursos de cómputo y de almacenamiento, además resulta especialmente apto para la explotación comercial de las grandes capacidades de cómputo de proveedores de servicios en Internet. (3)

### 2.2.3. Clasificación de soluciones Cloud Computing

Las soluciones de *cloud computing* disponibles en el mercado en la actualidad admiten diferentes clasificaciones según el aspecto que se tenga en cuenta

En base a la documentación analizada y tomando como referencias principales los informes del NIST (*NIST Cloud Computing Standards Roadmap*) y Deloitte (*Cloud Computing: Forecasting change. Market Overview and Perspective*) se definen tres características fundamentales que marcan la clasificación de las soluciones *cloud*: familias, formas de implementación y agentes intervinientes.

Estas tres características, junto con sus diferentes tipos de soluciones asociadas, se pueden representar en un cubo de tres dimensiones, tal y como se muestra en la figura inferior:

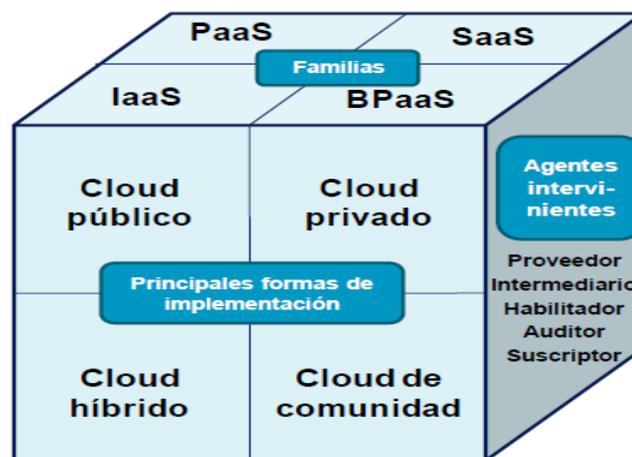


Figura II - 4. Cubo Cloud Computing

Mediante la combinación de estas tres dimensiones se detalla los distintos tipos de *cloud computing* existentes en el mercado así como sus principales agentes.

### **2.2.3.1. Modelos de Servicio**

#### **2.2.3.1.1. Infrastructure as a Service (IaaS)**

Familia de *cloud computing* consiste en poner a disposición del cliente el uso de la infraestructura informática (capacidad de computación, espacio de disco, memoria entre otros) como un servicio.

Los clientes que optan por este tipo de familia *cloud* en vez de adquirir o dotarse directamente de recursos como pueden ser los servidores, el espacio del centro de datos o los equipos de red optan por la externalización en busca de un ahorro en la inversión en sistemas TI.

#### **2.2.3.1.2. Platform as a Service (PaaS)**

Familia de *cloud computing* consiste en la entrega como un servicio de un conjunto de plataformas informáticas orientadas al desarrollo, testeo, despliegue, hosting y mantenimiento de los sistemas operativos y aplicaciones propias del cliente.

Las principales características asociadas al Platform as a Service como solución cloud se exponen a continuación:

- ✓ Facilita el despliegue de las aplicaciones del cliente, sin el coste y la complejidad derivados de la compra y gestión del hardware y de las capas de software asociadas.
- ✓ Ofrece a través de redes de servicio IP todos los requisitos necesarios para crear y entregar servicios y aplicaciones web.

### **2.2.3.1.3. Software as a Service (SaaS)**

Familia de *cloud computing* consiste en la entrega de aplicaciones como servicio, siendo un modelo de despliegue de software mediante el cual el proveedor ofrece licencias de su aplicación a los clientes para su uso como servicio bajo demanda.

Los proveedores de los servicios SaaS pueden tener instalada la aplicación en sus propios servidores web permitiendo a los clientes acceder, (por ejemplo, mediante un navegador web), o descargar el software en los sistemas del contratante del servicio. En este último caso se producirá la desactivación de la aplicación una vez finalice el servicio o expire el contrato de licencia de uso.

La solución de *cloud computing* de Software as a Service puede estar orientada a distintos tipos de clientes según su condición:

- ✓ Usuarios particulares:
  - Servicios de ofimática en cloud.
  - Redes sociales.

- ✓ Red 2.0.
  - Usuarios profesionales
  - CRM
  - ERP

#### **2.2.3.1.4. Business Process as a Service (BPaaS)**

Familia de *cloud computing* consiste en la provisión como servicio de procesos de negocio end-to-end altamente estandarizados a través de su entrega dinámica, la modalidad de pago por uso y los modelos de consumo de autoservicio bajo demanda.

Su característica principal es que los recursos utilizados mediante esta solución para ejecutar los procesos de negocio son compartidos entre los diferentes clientes del proveedor o administrador. En muchos casos este hecho proporciona un aporte de valor al negocio; sin embargo, la solución BPaaS se encuentra fase incipiente, siendo todavía un modelo de negocio en el que los proveedores tan solo operan en la actualidad en nichos concretos.

#### **2.2.3.2. Tipos de Cloud Computing**

##### **2.2.3.2.1. Cloud Público (Externo)**

Forma de implementación caracterizada por la oferta de servicios de computación virtualizados (bases de datos, sistemas operativos, plataformas de desarrollo,

aplicaciones, etc.) por parte de los proveedores para múltiples clientes, accediendo éstos a servicios a través de Internet o redes privadas virtuales (VPNs).

Como características inherentes a esta forma de implementación podemos citar las siguientes:

- ✓ Reducido plazo de tiempo para la disponibilidad del servicio.
- ✓ No se requiere llevar a cabo inversión monetaria para su implementación.
- ✓ Permite la externalización a un proveedor de servicios *cloud* de todas las funciones básicas de la empresa.
- ✓ Posibilita el aprovechamiento de la infraestructura de los proveedores de servicios, permitiendo adicionalmente una alta escalabilidad y flexibilidad en la modificación del dimensionamiento del servicio.
- ✓ Favorece la utilización de conjuntos de software estándar.
- ✓ Lleva asociadas unas cuotas iniciales de pago más bajas que el resto de implementaciones.
- ✓ Adicionalmente los costes del *cloud* público son variables, cumpliendo el principio de pago por uso.

La información corporativa se encuentra alojada en la nube pública junto a la del resto de clientes del proveedor, lo que implica, además de no poder tener localizada físicamente dicha información, imponer al proveedor una serie de requisitos de alta exigencia en temas de seguridad y protección de datos.

#### 2.2.3.2.2. Cloud Privado (Interno)

Forma de implementación caracterizada por el suministro por parte del proveedor o administrador, de entornos virtualizados que pueden ser implementados, usados y controlados por la misma empresa contratante del servicio. Esto indica no solo que la solución *cloud* puede ser administrada por la organización contratante, por el proveedor o por un tercer actor; sino que puede existir en las instalaciones propias del cliente o fuera de las mismas<sup>4</sup>.

Como características propias de esta forma de implementación se enumeran las siguientes:

- ✓ Reducido plazo de tiempo para la puesta en servicio y una alta flexibilidad en la asignación de recursos.
- ✓ Al contrario que el *cloud público*, requiere de inversión económica para la implementación de la solución contratada.
- ✓ Ofrece la posibilidad de aprovechar el personal existente y las inversiones en sistemas de información realizadas con anterioridad.
- ✓ Implica más especificidad en la solución adquirida, ya que está diseñada para ajustarse a las necesidades propias de la empresa contratante.
- ✓ Lleva asociados sistemas y bases de datos locales.

---

<sup>4</sup> Federal *Cloud Computing Strategy*, Vivek Kundra. (<http://www.cio.gov/documents/federal-cloud-computing-strategy.pdf>)

- ✓ Permite disponer de un control total de la infraestructura, de los sistemas y de la información corporativa tratada por éstos.
- ✓ Facilita el control y la supervisión de los requisitos de seguridad y protección de la información almacenada.

#### **2.2.3.2.3. Cloud de Comunidad**

Se trata de *clouds* utilizados por distintas organizaciones cuyas funciones y servicios sean comunes, permitiendo con ello la colaboración entre grupos de interés.

Ejemplos de esta forma de implementación son los *clouds* de comunidades de servicios de salud (en inglés, *healthcare community cloud*) para facilitar el acceso aplicaciones e información crítica de carácter sanitario y los *clouds* de comunidad gubernamentales (en inglés, *government community cloud*) para facilitar el acceso a recursos de interoperabilidad entre organismos públicos y administraciones públicas.

Al analizar un *cloud* de comunidad se debe considerar que en un principio sus fortalezas y debilidades se sitúan entre las del privado y las del público. En general el conjunto de recursos disponibles con un *cloud* de comunidad es mayor que en el privado con las ventajas evidentes que conlleva en términos de elasticidad. Sin embargo, la cantidad de recursos es menor que los existentes en una solución de *cloud* público, limitando la elasticidad respecto a dicho *cloud* público.

Por otra parte, el número de usuarios de este tipo de nube es menor que los de la nube pública, lo que la dota de mayores prestaciones en cuestiones de seguridad y privacidad<sup>5</sup>.

#### **2.2.3.2.4. Cloud Híbrido**

Forma de implementación cuya infraestructura *cloud* (en la nube) se caracteriza por aunar dos o más formas de *clouds* (privado, comunitario o público), los cuales continúan siendo entidades únicas interconectadas mediante tecnología estandarizada o propietaria, que permite la portabilidad de datos y aplicaciones (ej. el rebalanceo de cargas entre nubes).

Una entidad que emplee esta forma de implementación se podría beneficiar de las ventajas asociadas a cada tipo de *cloud*, disponiendo con ello de una serie de características adicionales tal y como se muestra a continuación:

- ✓ Ofrece una mayor flexibilidad en la prestación de servicios de TI, al mismo tiempo que se mantiene un mayor control sobre los servicios de negocio y de datos.
- ✓ Con una solución de *cloud* híbrido, al igual que en los casos detallados anteriormente se consigue una rápida puesta en servicio.

---

<sup>5</sup> Security & Resilience in Governmental Clouds, ENISA. (<http://www.enisa.europa.eu/act/rm/emerging-and-future-risk/deliverables/security-and-resilience-in-governmental-clouds>)

- ✓ Implica mayor complejidad en la integración de la solución *cloud*, como consecuencia de ser una solución que se compone de dos formas distintas de implementación de servicios en la nube.
- ✓ Permite integrar las mejores características de las dos formas de implementación *cloud*, en cuanto al control de los datos y a la gestión de las funciones básicas de la entidad.
- ✓ Posibilita la selección por parte del proveedor, de infraestructura escalable y flexible permitiendo una alta agilidad en el redimensionamiento de la solución.
- ✓ Permite el control interno de los servicios *cloud* desde la propia entidad.

### **2.3. Hipervisor**

La virtualización se refiere a la abstracción de los recursos de una computadora llamada Hipervisor o VMM en sus siglas en inglés (Virtual Machine Monitor) que crea una capa de abstracción entre el hardware de la máquina física (host) y el sistema operativo de la máquina virtual (virtual machine, guest), siendo un medio para crear una versión virtual de un dispositivo o recurso como un servidor, un dispositivo de almacenamiento, una red o incluso un sistema operativo donde se divide el recurso en uno o más entornos de ejecución.<sup>6</sup> (4)

---

<sup>6</sup> <http://es.wikipedia.org/wiki/Virtualizaci%C3%B3n>



**Figura II - 5. Arquitectura de Trabajo de un Hipervisor**

Esta plataforma de virtualización se encarga de gestionar los cuatro recursos principales de un computador (CPU Central Processing Unit, Unidad Central de Procesamiento, memoria, red y discos de almacenamiento) repartiendo dinámicamente dichos recursos entre todas las VMs definidas en el computador anfitrión.

### **2.3.1. Tipo de Hipervisores**

Existen dos tipos fundamentales de Hipervisores (13)

#### **2.3.1.1. Hipervisores de Tipo 1 o Bare-Metal**

Capaces de ejecutarse directamente sobre el hardware real de la máquina sin necesidad de tener instalado ningún SO. Este tipo de Hipervisores son más eficientes que los de tipo 2, ya que consiguen dar un mayor rendimiento y escalabilidad, además de suponer

menos sobrecarga para el computador. La desventaja es que no todo el hardware soporta este tipo de software. Ejemplos de este tipo de hipervisores son Xen , KVM

#### **2.3.1.1.1. XEN**

Xen es una plataforma software de virtualización. Se trata de un producto software bajo licencia GPL v2, desarrollado bajo una aproximación comunitaria con la participación de grandes empresas como AMD, Cisco, Dell, Fujitsu, HP, IBM, Intel, Mellanox, Network Appliance, Novell, Red Hat, Samsung, SGI, Sun, Unisys, Veritas, Voltaire, y Citrix. Esto lo convierte en un estándar industrial con soporte generalizado. Como características técnicas destacables, Xen soporta paravirtualización.

El proyecto Xen ha ido incorporando subproyectos para abordar otros elementos de la arquitectura cloud. Sobre estos conviene destacar la plataforma Xen Cloud donde se está acometiendo el desarrollo de toda la arquitectura software para el desarrollo de clouds. (15)

#### **2.3.1.1.2. KVM**

KVM es una plataforma software de virtualización. Se trata de un producto software bajo licencia LGPL, desarrollado dentro del RedHat Emerging Technology. Como características técnicas destacadas, KVM soporta arquitecturas Intel, AMD y sistemas operativos Windows, Linux y Unix, admite multiproceso simétrico de máquinas virtuales Tecnologías / Proveedores, migración hardware en caliente implementa información de estado de las máquinas virtualizadas, de los host y admite

paravirtualización de la red y los dispositivos de bloque. Numerosas empresas ofrecen soporte comercial de KVM. (10)

### 2.3.1.2. Hipervisores de Tipo 2 o Hosted

Necesitan un SO anfitrión para poder ejecutarse. Permiten la creación de VMs dentro del mismo SO. Al ejecutarse sobre un SO y no directamente sobre el hardware, el rendimiento de este tipo de hipervisores es menor que los de tipo 1. Suelen utilizarse en entornos de escritorio y no en grandes infraestructuras virtualizadas. Algunos de ellos son VirtualBox , VMware Player o Virtual PC.

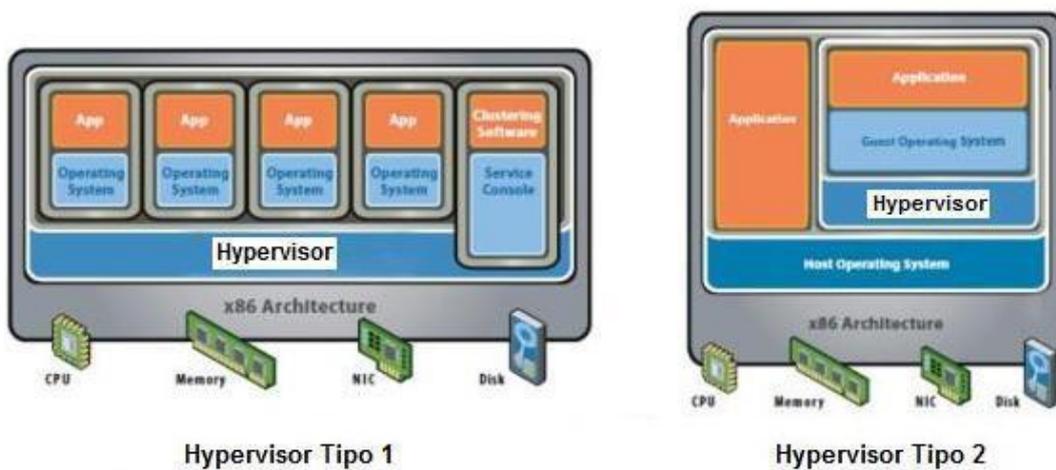


Figura II - 6. Tipo de Hipervisores

## **CAPÍTULO III**

### **Análisis Comparativo de las Tecnologías de Infraestructura como Servicio Open Source para Nubes Privadas en Cloud Computing**

#### **3.1. Introducción**

La determinación de una tecnología de Infraestructura como Servicio Open Source para Nubes Privadas en Cloud Computing debe ser una de las decisiones más importantes, la misma debe estar basada en un profundo y minucioso análisis de acuerdo a criterios o parámetros de comparación.

#### **3.2. Determinación de las Tecnologías a comparar**

En la actualidad existen en el mercado una gama variada de tecnologías de Infraestructura como Servicio que son bastante populares y de gran importancia, de las cuales se ha seleccionado tres tecnologías por las razones que a continuación se explican:

✓ La siguiente página<sup>7</sup> describe: “Los paquetes de fuente abierta adoptan un enfoque holístico, integrando toda la funcionalidad necesaria en un solo paquete (incluyendo la virtualización, gestión, interfaces y seguridad). Cuando se añade a una red de servidores y almacenamiento, estos paquetes producen el *cloud computing* flexible y las infraestructuras de almacenamiento (IaaS).”

- Eucalyptus
- Nimbus
- OpenNebula

✓ En el siguiente portal<sup>8</sup> se destaca: El software libre ha ido en aumento en muchas empresas durante la recesión económica prolongada y una de las áreas que empieza a ofrecer a las empresas una gran cantidad de flexibilidad y ahorro de costes es el *cloud computing*. Implementaciones de nube pueden ahorrar dinero.

- Digg
- Eucalyptus
- Red Hat's Cloud
- Traffic Server
- Cloudera

---

<sup>7</sup> <http://www.ibm.com/developerworks/opensource/library/os-cloud-anatomy/>

<sup>8</sup> <http://www.cloudssky.com/en/openiaas/open-iaas-solutions.html>

- Puppet
- Enomaly
- Joyent
- Zoho
- Nimbus
- Reservoir
- OpenNebula

✓ El siguiente portal<sup>9</sup> describe: “Lista pequeña para conocer plataformas cloud (IaaS y PaaS):” de las cuales solo tomamos las IaaS

- Eucalyptus
- OpenNebula
- CloudStack

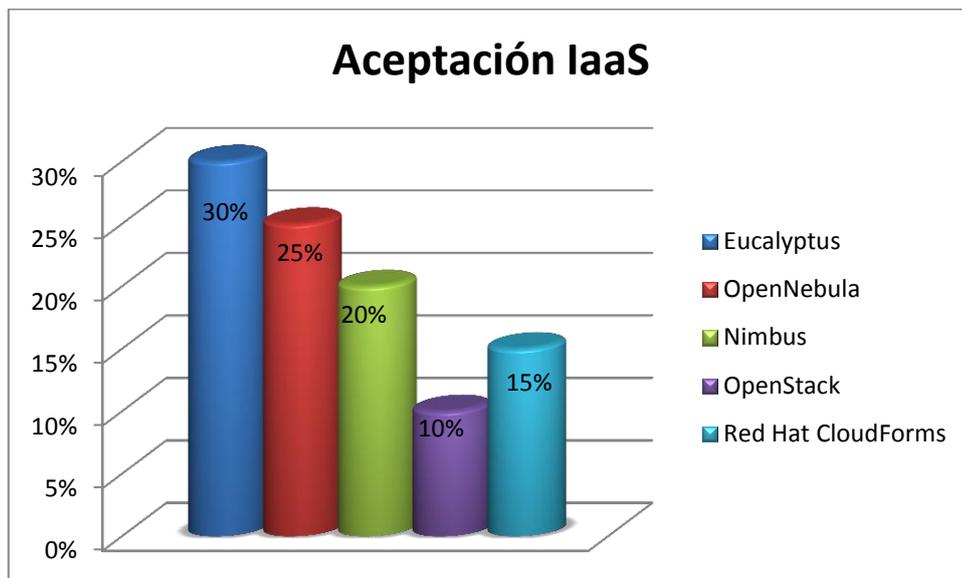
✓ La encuesta realizada por medio de un entorno muy práctico como es Google Documents a personal técnico en empresas , organismos e instituciones educativas de nivel superior tanto públicas como privadas sobre cual de las siguientes IaaS Open Source tienen mayor aceptación-funcionalidad arrojaron los siguientes datos.

- Eucalyptus
- OpenNebula

---

<sup>9</sup> <http://www.coloandcloud.com/editorial/open-source-cloud-platforms-iaas-paas/>

- Nimbus
- OpenStack
- Ret Hat Cloud Forms



**Figura III - 7. Resultados Encuesta Aceptación IaaS**

Se seleccionó tres tecnologías de IaaS para la realización del análisis comparativo:

Eucalyptus, Nimbus, y OpenNebula de acuerdo a las siguientes consideraciones:

- ✓ En varias páginas web consultadas las tres IaaS seleccionadas gozan de una gran popularidad.
- ✓ Analizada información de cada una de ellas cuentan con una amplia gama de información acerca de su configuración y administración.
- ✓ Los resultados proyectados de la encuesta realizada por medio de Google Documents determinaron que las IaaS seleccionadas poseen alto nivel de aceptación y funcionalidad para los técnicos de las diversas organizaciones públicas y privadas.

- ✓ La optimización de recursos tanto en tecnología como en lo económico es una de las mayores ventajas para preferir las IaaS Open Source.
- ✓ La facilidad y la flexibilidad que las IaaS proporcionan para el manejo y administración de recursos computacionales es realmente importante.

### **3.3. Análisis de las Tecnologías de Infraestructura como Servicio seleccionadas**

#### **3.3.1. Eucalyptus**



**Figura III - 8. Logo de Eucalyptus**

##### **3.3.1.1. Generalidades**

Eucalyptus es una arquitectura software open source basada en Linux que implementa clouds privados e híbridos dentro de una infraestructura de TI de una organización. Inicialmente fue diseñado y desarrollado por el equipo del profesor Rich Wolski, como un proyecto del Computer Science Department de la Universidad de California. Eucalyptus proporciona la infraestructura como un servicio (IaaS) de tal forma que los usuarios pueden provisionar sus propios recursos (hardware, almacenamiento y red) en función de sus necesidades. Se diseñó para ser fácil de instalar de forma menos intrusiva posible.

Eucalyptus proporciona una capa de red virtual de tal forma que se aísla el tráfico de red de diferentes usuarios y permite que uno o más clusters parezcan pertenecer a la misma

LAN (sólo Enterprise Edition). Además, tiene la capacidad de interactuar con Amazon EC2 y los servicios S3 de cloud público ofreciendo la posibilidad de crear un cloud híbrido. (2)

Las funcionalidades principales de la plataforma cloud de Eucalyptus son:

- ✓ API compatible con AWS
- ✓ Arquitectura agnóstica en relación al hipervisor (soporta Xen, KVM, VMware)
- ✓ Gestor Walrus de almacenamiento compatible con S3 (gestiona el almacenamiento de los datos de usuario, así como las imágenes de los sistemas de archivos de las máquinas virtuales, los kernels y los ramdisks) Soporte EBS (Elastic Block Store) sobre AoE e iSCSI
- ✓ Múltiples modos de funcionamiento de red para adaptarse a diferentes arquitecturas
- ✓ Interfaz Web y herramientas CLI para la administración y la configuración del cloud
- ✓ Arquitectura escalable: las peticiones del cloud sirven de forma asíncrona. Los usuarios de Eucalyptus que interactúan con el cloud disponen de diferentes funciones para implementar, gestionar y mantener sus propias colecciones de recursos virtuales.

Entre ellas se incluye:

- ✓ Gestión de claves SSH. Se emplean claves privadas y públicas para validar la identidad del usuario cuando se conecta a las máquinas virtuales por medio de

SSH. Los usuarios de Eucalyptus pueden añadir, obtener información y borrar pares de claves.

- ✓ Gestión de imágenes. Antes de poder ejecutar una máquina virtual, éstas deben ser preparadas para poder emplearse en el cloud. Los usuarios de Eucalyptus pueden empaquetar, subir, registrar, obtener información, descargar, desempaquetar y cancelar el registro de las imágenes de las máquinas virtuales.
- ✓ Gestión de máquinas virtuales. Los usuarios pueden ejecutar sus propias máquinas virtuales en el cloud. Los usuarios pueden ejecutar, obtener información, apagar y reiniciar una gran variedad de máquinas virtuales Linux que fueron preparadas empleando las funciones de gestión de imágenes de Eucalyptus.
- ✓ Gestión de direcciones IP. Dependiendo del modo de gestión de red establecido, los usuarios podrán tener acceso al sistema de IPs elásticas, lo que les permitirá asignar, asociar, obtener información y liberar direcciones IP.
- ✓ Gestión de grupos. “Security groups” son conjuntos de reglas de firewall aplicadas a las instancias de las máquinas virtuales asociadas a un grupo. Eucalyptus permite crear, obtener información, borrar, autorizar y revocar grupos de seguridad.
- ✓ Gestión de volúmenes y Snapshot. Eucalyptus permite a los usuarios crear bloques de volúmenes de forma dinámica que podrán ser empleados con las máquinas virtuales. Los usuarios pueden crear, asignar, liberar, obtener información y borrar volúmenes. Los usuarios pueden crear y borrar snapshots y crear nuevos volúmenes.

### **3.3.1.2. Arquitectura**

#### **3.3.1.2.1. Interna**

La arquitectura de Eucalyptus es muy modular y sus componentes internos emplean servicios Web, lo que hace factible reemplazarlos con facilidad e incluso expandirlos. Eucalyptus implementa el API Amazon Web Service (AWS) que permite la interoperabilidad con los servicios existentes compatibles con AWS. Esto permite que los usuarios puedan agrupar recursos tanto de un cloud privado interno como de uno externo para formar un cloud híbrido.

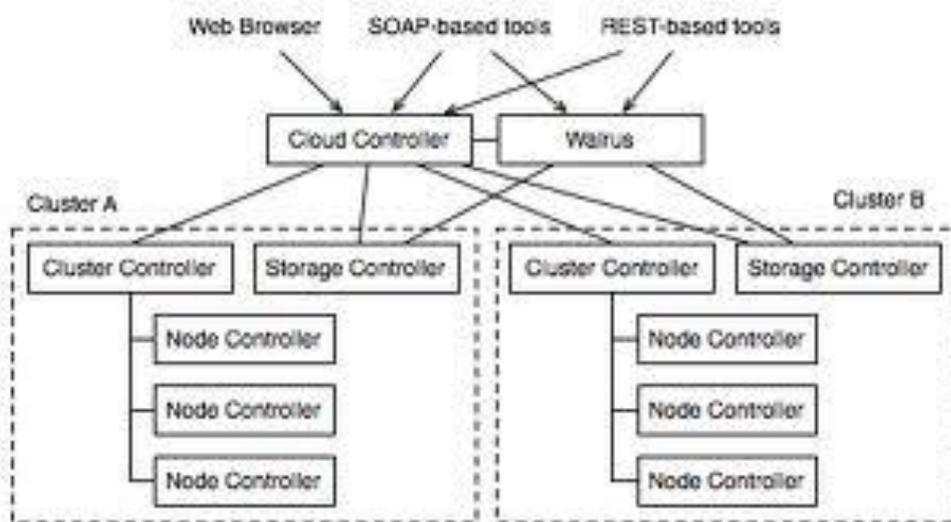
#### **3.3.1.2.2. Infraestructura**

El cloud de Eucalyptus consta de cinco tipos de componentes. El cloud controller (CLC) y Walrus son componentes de alto nivel. En una instalación de cloud debe haber ambos. El cloud controller es un programa en Java que ofrece interfaces SOAP compatibles con EC2, así como una interfaz Web para todo el mundo. Además de gestionar las peticiones entrantes, el cloud controller aporta un planificador de recursos de alto nivel y un sistema de contabilidad. Walrus, también está escrito en Java, implementa almacenamiento bucket-based, el cual estaría disponible fuera y dentro del cloud mediante interfaces REST y SOAP compatibles con S3.

Los componentes de alto nivel pueden agregar recursos de múltiples clusters (conjuntos de nodos compartiendo un segmento de LAN, posiblemente ubicados detrás de un firewall). Cada cluster necesita un cluster controller (CC) para efectuar las tareas de planificación y control de la red, así como un storage controller (SC), para el almacenamiento basado en bloques EBS. Los componentes de segundo nivel deberían

ser desplegados en el nodo cabecera de cada cluster. Finalmente, cada nodo necesitará un node controller (NC) para controlar el hipervisor que se encuentra instalado en el mismo. El CC y el NC están escritos en C y se despliegan como servicios Web dentro de Apache. El SC está escrito en Java. La comunicación entre esos componentes tiene lugar por medio de SOAP con seguridad WS.

A continuación una descripción detallada de cada uno de los componentes de Eucalyptus:



**Figura III - 9. Arquitectura IaaS Eucalyptus**

### **Cloud controller (CLC)**

El Cloud Controller proporciona una interfaz compatible con los servicios Web EC2/S3. Además una interfaz Web para los usuarios, que permite gestionar ciertos aspectos de la infraestructura.

### **Funciones:**

- ✓ Monitoriza la disponibilidad de recursos en varios componentes de la infraestructura, incluyendo los nodos y los Cluster Controllers

- ✓ Efectúa el arbitraje de recursos: decide qué clusters se emplearán para provisionar las instancias
- ✓ Monitoriza las instancias en ejecución

### **Walrus**

Implementa el sistema de almacenamiento persistente empleando APIs REST y SOAP compatibles con S3

#### **Funciones:**

- ✓ Almacena las imágenes maestras de las máquinas virtuales
- ✓ Almacena snapshots
- ✓ Almacena y sirve los archivos empleando el API S3

### **Cluster controller (CC)**

Efectúa las tareas de planificación y control de la red a nivel de cluster

### **Storage controller (SC)**

Gestiona el almacenamiento EBS a nivel de cluster

#### **Funciones:**

- ✓ Gestiona la creación de los dispositivos EBS
- ✓ Proporciona a las instancias el acceso al almacenamiento por bloques mediante AoE o iSCSI
- ✓ Permite la creación de snapshots de los volúmenes

### **Node controller (NC)**

Se ejecuta en cada uno de los nodos y gestiona el ciclo de vida de las instancias.

Interactúa con el hipervisor y con el Cluster Controller

#### **Funciones:**

- ✓ Gestiona el ciclo de vida de las instancias

- ✓ Recolecta los datos relacionados con la disponibilidad de los recursos e informa al Cluster Controller.

### 3.3.2. Nimbus



**Figura III - 10. Logo Nimbus**

#### 3.3.2.1. Generalidades

Nimbus, es un conjunto de herramientas open source que proveen un modelo IaaS. El proyecto inició en el año 2007 en la Universidad de Chicago y está en producción desde marzo del año 2008. Actualmente es un proyecto incubadora del Globus Alliance, soportado por la National Science Foundation - NSF y el Department of Energy - DOE de los Estados Unidos. El objetivo principal de Nimbus es la provisión de infraestructuras computacionales para el soporte de proyectos de la comunidad científica, con el fin de mejorar el entendimiento de las potencialidades y desafíos del paradigma *cloud computing*.

Entre sus principales funcionalidades, Nimbus permite el despliegue con clusters virtuales a gran escala en forma rápida, automática y repetitiva. Esto incluye la provisión de mecanismos para facilitar la personalización de instancias de máquinas virtuales a través de un script ejecutado al iniciar las máquinas virtuales, esto es configuración de políticas de acceso y configuraciones de almacenamiento sobre una máquina virtual, adaptándola al contexto deseado por el usuario final. Estos clusters

virtuales también pueden ser configurados para pertenecer a una red privada virtual, de tal forma que se facilite la ejecución de aplicaciones con requerimientos de interconexión entre instancias. (11)

### 3.3.2.2. Arquitectura

#### 3.3.2.2.1. Interna

La arquitectura de Nimbus es modular cuyo objetivo es proporcionar a las funciones administrativas y de usuario de manera amigable. Se centra alrededor de una aplicación Python que está destinada a ser de despliegue del servicio Nimbus.

#### 3.3.2.2.2. Infraestructura

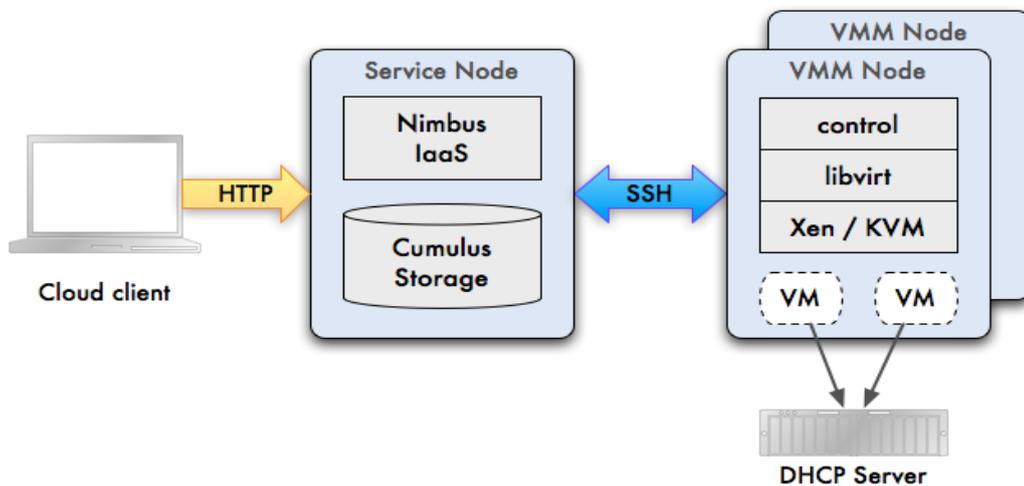


Figura III - 11. Arquitectura IaaS Nimbus

#### Workspace Service

Es un administrador del sitio VM independiente que pueden invocar diferentes interfaces de protocolo remotos.

Los protocolos soportados actualmente son los basados en Servicios Web o en HTTP. Todos corren, ya sea en un Apache Axis basado Java contenedor o Apache CXF . Pero sólo hay un cierto nivel de necesidad:

- ✓ No hay nada específico a los servicios web basados en protocolos remotos en la implementación del workspace service, el sistema de mensajería sólo tiene que ser capaz de hablar a las bibliotecas basadas en Java.
- ✓ Las dependencias workspace service no tienen nada que ver con el envase que se está ejecutando, son normales las dependencias de aplicaciones Java como Spring , ehcahe , backport-util concurrente y JDBC (actualmente usando el incorporado Derby base de datos).

### **Cumulus**

Es una implementación de código abierto del API REST S3.

Es el Front End para el repositorio de imágenes de máquinas virtuales de Nimbus

En orden de arranque de una imagen nimbus dada, primero debe estar en el repositorio Cumulus.

### **Workspace Pilot**

Es un programa administrador de recursos locales (LRM) a fin de obtener el tiempo en los nodos de VMM. Cuando no está atribuida al Workpace service, estos nodos se utilizará para los trabajos de forma normal (los trabajos se ejecutan en las cuentas normales del sistema de dominio Xen 0 sin máquinas virtuales invitadas en funcionamiento).

Varias medidas de seguridad adicionales se han añadido para asegurar que el nodo devuelto por VM modo de hospedaje en el momento adecuado, incluyendo soporte para:

- ✓ Workspace service de estar abajo o mal funcionamiento
- ✓ Preferencia LRM (incluida la cancelación deliberado trabajo LRM)
- ✓ Nodo reinicio / apagado

También se incluye un solo comando "kill 9" facilidad para los administradores como un "peor escenario" contingencia.

El uso de pilot es opcional. Por defecto el servicio no funciona con ello, el servicio gestiona directamente los nodos que están configurados para manejar.

### **Workspace-control**

Programa instalado en cada nodo VMM se usa para:

- ✓ Iniciar, detener y pausar VMs,
- ✓ Implementar reconstrucción VM y administración de imágenes,
- ✓ Conectan de forma segura de las máquinas virtuales a la red, y
- ✓ Entregar información contextualización.

En la actualidad, las herramientas de workspace-control trabajan con Xen y KVM.

Implementado en Python para ser portátil y fácil de instalar. Requiere sudo libvirt, ebtables y una biblioteca de servidor DHCP.

### 3.3.3. OpenNebula



Figura III - 12. Logo OpenNebula

#### 3.3.3.1. Generalidades

OpenNebula es un software open-source que permite construir cualquier tipo de cloud privado, público e híbrido. Ha sido diseñado para ser integrado con cualquier tipo de red y almacenamiento, para así adaptarse a cualquier centro de datos existente.

OpenNebula gestiona el almacenamiento, las redes y las tecnologías de virtualización. Proporciona la posibilidad de desplegar servicios en infraestructuras distribuidas, combinando recursos de centros de datos así como de clouds remotos, de acuerdo con las políticas de despliegue. (12)

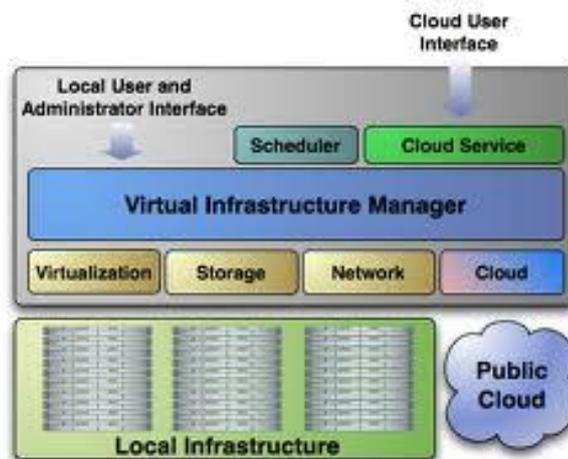


Figura III - 13. Capas OpenNebula

### 3.3.3.2. Arquitectura

#### 3.3.3.2.1. Interna

La arquitectura interna de OpenNebula se divide en tres capas:

##### Tools

Herramientas de gestión desarrolladas empleando las interfaces proporcionadas por el núcleo de OpenNebula

##### Core

Componentes principales para gestionar las máquinas virtuales, redes virtuales y nodos

##### Drivers

Proporcionan nuevas tecnologías para la virtualización, el almacenamiento, la monitorización y los servicios de cloud

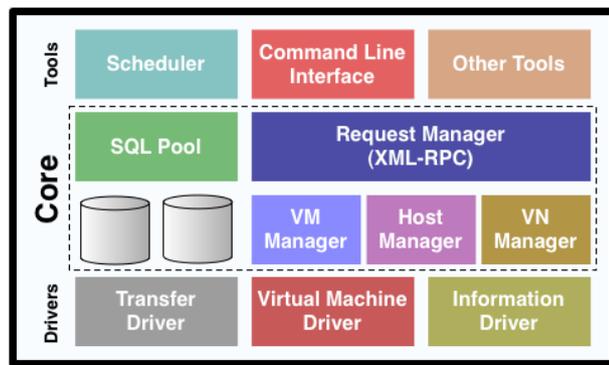


Figura III - 14. Arquitectura IaaS OpenNebula

##### Tools

Esta capa contiene las herramientas que se distribuyen con OpenNebula, como el CLI, el planificador, el API libvirt, así como herramientas desarrolladas por terceras partes que emplean el interfaz XML-RPC de OpenNebula:

### **Command Line Interface**

Permite a los usuarios y administradores de OpenNebula gestionar de forma manual la infraestructura virtual.

### **Scheduler**

Es una entidad independiente en la arquitectura y puede desacoplarse del resto de los componentes. Emplea la interfaz XML-RPC para invocar las acciones que se efectuarán en las máquinas virtuales.

## **OpenNebula Core**

Está compuesto por un conjunto de componentes que permiten gestionar y monitorizar máquinas virtuales, redes virtuales, almacenamiento y nodos. Los componentes principales del núcleo son:

### **Request Manager**

Gestiona las peticiones de los clientes

### **Virtual Machine Manager**

Gestiona y monitoriza las máquinas virtuales

### **Transfer Manager**

Gestiona la transferencia de imágenes

### **Virtual Network Manager**

Gestiona las redes virtuales

### **Host Manager**

Gestiona y monitoriza los recursos físicos

### **Database**

Base de datos persistente que almacena los datos de OpenNebula. Soporta SQLite3 y MySQL

### Driver Modules

Interactúan con middleware específico (hipervisor, servicios de cloud, transferencia de imágenes, o servicios de información)

### 3.3.3.2.2. Infraestructura

OpenNebula emplea en su infraestructura una arquitectura en cluster clásica, con un frontal y con conjunto de nodos donde serán ejecutadas las máquinas virtuales. Al menos, debe haber una red que interconecte todos los nodos con el frontal

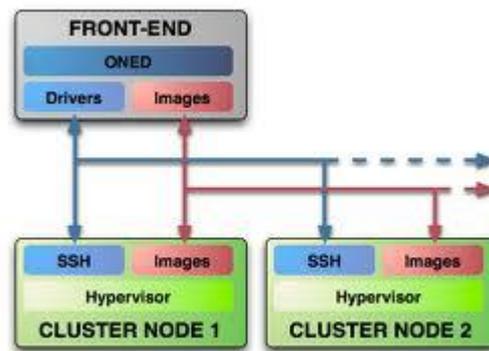


Figura III - 15. Componentes OpenNebula

Los componentes de OpenNebula son:

#### Front-end

Ejecuta OpenNebula y los servicios del cluster

#### Nodos

Anfitriones que proporcionan los recursos necesarios a las máquinas virtuales, como el software hipervisor

#### Repositorio de imágenes

Medio que almacena las imágenes virtuales base

### **Demonio OpenNebula**

Gestiona el ciclo de vida de las máquinas virtuales y los subsistemas (red, almacenamiento, hipervisores)

### **Drivers**

Programas empleados por el núcleo de OpenNebula para servir de interfaz para un hipervisor o un sistema de almacenamiento específico

### **Tipos de usuarios:**

#### **oneadmin**

Usuario administrador del cloud privado que gestiona las máquinas virtuales, redes, nodos o usuarios

#### **Usuarios no privilegiados**

Gestionan única y exclusivamente sus propios objetos (máquinas virtuales, redes virtuales). Hay que señalar que pueden instanciar aquellas imágenes virtuales del repositorio a las que se les ha establecido el acceso público

### **3.4. Determinación de parámetros de comparación**

Para la realización del análisis comparativo de Tecnologías de Infraestructuras como Servicio Open Source para Nubes Privadas en Cloud Computing, se debe considerar una serie de aspectos, examinando las características más relevantes de cada infraestructura. Las tres Infraestructuras como Servicio (Eucalyptus, Nimbus y Opennebula) poseen similares funcionalidades pero difieren en la forma como realizan sus operaciones de despliegue de servicios.

Se ha establecido los siguientes parámetros cada uno con sus indicadores para determinar por separado las potencialidades y debilidades que las IaaS poseen. A continuación se describe los parámetros de comparación.

#### **3.4.1. Configuración**

Parámetro que determina indicadores que permiten medir el número de pasos necesarios, el conocimiento sobre diferentes aspectos técnicos, la utilización de tiempo de trabajo y recurso humano.

#### **3.4.2. Flexibilidad**

Parámetro que establece indicadores que permiten conocer la cantidad de soporte a Hipervisores, uso de DHCP, modelo de despliegue de servicio, entornos de programación en los cuales ha sido desarrollada la IaaS.

#### **3.4.3. Gestión de Usuarios**

Parámetro que establece indicadores para determinar la creación, eliminación, modificación, monitoreo de usuarios, así como el protocolo de seguridad que usa para la autenticación de los usuarios al momento de iniciar el despliegue de servicios computacionales ofrecidos por la IaaS.

#### **3.4.4. Gestión de Nodos**

Parámetro que estable indicadores para determinar el grado de administración configuración y monitoreo de los nodos que sirven de base para el despliegue de servicios dentro de la IaaS.

#### **3.4.5. Gestión de Máquina Virtuales**

Parámetro que estable indicadores para determinar el grado de administración, configuración y monitoreo de servicio (máquinas virtuales) de la IaaS así como la variedad de sistemas operativos soportados por el servicio.

Cada parámetro tiene un valor porcentual que se le asignó bajo los diversos criterios.

Se ha asignado mayor porcentaje al parámetro Gestión de Máquinas Virtuales ya que es el pilar fundamental en el que se basa la entrega de servicios que brinda la IaaS, parámetros como: Configuración, Flexibilidad, Gestión de Nodos tienen un porcentaje medio debido a que son ejes necesarios para iniciar con la entrega de los servicios, el parámetro Gestión de Usuarios tiene un porcentaje menor ya que indica hacia quienes se dirige el servicio.

El porcentaje de peso de los parámetros que se van a considerar en general en este análisis son los siguientes:

**Tabla III. I Porcentaje de pesos de los parámetros de evaluación**

Parámetros	Justificación	%
Configuración	Este parámetro es la base para la determinar niveles de funcionalidad y administración de la IaaS por lo que se determinó el siguiente porcentaje	20%
Flexibilidad	Este parámetro sirve para conocer la abstracción que la IaaS presenta por lo que se estableció el siguiente porcentaje.	20%
Gestión de Usuarios	Este parámetro orienta a hacia quienes se proveerá el servicio que brinda la IaaS se le determinó un porcentaje menor sin que esto estipule la importancia que este tiene para el análisis.	10%
Gestión de Nodos	Este parámetro sirve de apoyo para el despliegue de servicios dentro de la IaaS por lo que se determinó el siguiente porcentaje.	20%
Gestión de Máquinas Virtuales	Este parámetro es el pilar fundamental para la IaaS se enmarca en la entrega de servicios computacionales por lo que se ha determinado el mayor porcentaje.	30%
<b>TOTAL</b>		<b>100%</b>

Fuente: Roberto Carlos Salazar Gualoto

Los parámetros de evaluación que se ha tomado en cuenta para desarrollar el análisis comparativo se dividen en indicadores que se detallan a continuación:

### 3.4.6. Indicadores parámetro Configuración

**Tabla III. II Indicador Grado de Complejidad**

Indicador		Descripción	
Grado de complejidad		Es el grado de complejidad que requieren los pasos necesarios para la implementación inicial de la IaaS	
<b>Forma de medición Cuantitativa según el número de pasos o comandos a ejecutar</b>			
Muy Bajo	Bajo	Medio	Alto
<=50	>50 y <=100	>100 y <= 150	>=200

Fuente: Roberto Carlos Salazar Gualoto

**Tabla III. III Indicador Nivel de Conocimiento**

Indicador		Descripción	
<b>Nivel de Conocimiento</b>		<b>Es el nivel de conocimiento que se necesita para realizar la configuración inicial de la IaaS</b>	
<b>Índices de Conocimientos necesarios</b>			
Instalación Básica de Linux Herramientas Básicas de Configuración Administración y Utilización de Paquetes YUM y RPM Creación y Eliminación de Directorios Configuración de Gestores de Arranque Administración Remota con SSH Extracción de Archivos Comprimidos TAR, UNZIP, GUNZIP Manejo de Permisos Editores de Texto VI Instalación y Administración de XEN Gestión de Usuarios y Contraseñas Herramienta de descarga de contenidos WGET Instalación de Python Servicio DHCP Configuración de Bridge Instalación de Ruby Delegación de Tareas Superusuario SUDO Manejo de Comandos Propios de Eucalyptus Manejo de Comandos Propios de Nmbus Manejo de Comandos Propios de OpenNebula			
<b>Forma de medición Cuantitativa según los índices de conocimiento</b>			
<b>Muy Bajo</b>	<b>Bajo</b>	<b>Medio</b>	<b>Alto</b>
≤4	>4 y ≤8	>8 y ≤ 12	>12

**Fuente:** Roberto Carlos Salazar Gualoto

**Tabla III. IV Indicador Cantidad de Tiempo**

Indicador		Descripción	
<b>Cantidad de Tiempo</b>		<b>Se refiere a la cantidad de horas de trabajo empleadas para la configuración inicial de la IaaS</b>	
<b>Forma de medición Cuantitativa según el cantidad de horas empleadas en la configuración</b>			
<b>Mala</b>	<b>Regular</b>	<b>Buena</b>	<b>Muy Buena</b>
≤0	>0 y ≤80	>40 y ≤ 80	>80

**Fuente:** Roberto Carlos Salazar Gualoto

**Tabla III. IV Indicador Cantidad de Recurso Humano**

Indicador		Descripción	
Cantidad de Recurso Humano		Se refiere a la cantidad de personal necesario para la implementación inicial de la IaaS con experiencia en diversas áreas de conocimiento técnicos indicados en la Tabla III. III	
Forma de medición Cuantitativa según el número de recurso humano necesario en la configuración			
Ninguno	Unos	Pocos	Muchos
<=0	>0 y <=2	>2 y <= 5	>5

Fuente: Roberto Carlos Salazar Gualoto

### 3.4.7. Indicadores parámetro Flexibilidad

**Tabla III. V Indicador Cantidad Soporte Hypervisor**

Indicador		Descripción	
Cantidad de Soporte Hypervisor		Se refiere al número de hypervisores que soporta la IaaS, son el eje principal del despliegue de los servicios.	
Forma de medición Cuantitativa según el número de hypervisores			
Ninguno	Unos	Pocos	Muchos
<=0	>0 y <=2	>2 y <= 5	>5

Fuente: Roberto Carlos Salazar Gualoto

**Tabla III. VI Indicador Grado de Uso de DHCP**

Indicador		Descripción	
Grado Uso de DHCP		Se refiere al grado de Uso de DHCP que presenta la IaaS es decir a la manera de solicitar Direcciones IP para el establecimiento de red de los servicios.	
Forma de medición Cuantitativa según la forma de uso de DHCP			
Muy Bajo	Bajo	Medio	Alto
No es necesario la implementación de DHCP	Requiere la implementación de DHCP estático	Requiere la implementación de DHCP automática	Requiere la implementación de DHCP dinámica

Fuente: Roberto Carlos Salazar Gualoto

**Tabla III. VII Indicador Nivel de Despliegue**

Indicador		Descripción	
Nivel de Despliegue		Se refiere al nivel utilizado por la IaaS para el despliegue de servicios	
Forma de medición Cuantitativa según la forma de despliegue de servicios			
Muy Bajo	Bajo	Medio	Alto
Si el despliegue es realizado de manera manual	Si el despliegue requiere ejecución de comandos para inicio de servicios	Si el despliegue es realizado de manera estática	Si el despliegue es realizado de manera dinámica

Fuente: Roberto Carlos Salazar Gualoto

**Tabla III. VIII Indicador Cantidad de Entorno de Programación**

Indicador		Descripción	
Cantidad de Entorno de Programación		Se refiere al número de entornos de programación en los que ha sido desarrollado las IaaS que permiten mayor abstracción de sus funcionalidades.	
Forma de medición Cuantitativa según el número de entornos de programación			
Ninguno	Unos	Pocos	Muchos
$\leq 0$	$>0$ y $\leq 2$	$>2$ y $\leq 5$	$>5$

Fuente: Roberto Carlos Salazar Gualoto

### 3.4.8. Indicadores parámetro Gestión de Usuarios

**Tabla III. IX Indicador Creación, Modificación, Eliminación de Usuarios**

Indicador		Descripción	
Creación, Modificación, Eliminación de Usuarios		Se refiere a los pasos que se realiza para crear modificar eliminar usuarios para la posterior entrega de servicios computacionales.	
Forma de medición Cuantitativa según el número de pasos			
Muy Complejo	Complejo	Simple	Muy Simple
$>20$	$>15$ y $\leq 20$	$>10$ y $\leq 15$	$\leq 10$

Fuente: Roberto Carlos Salazar Gualoto

**Tabla III. X Indicador Monitoreo de Usuarios**

Indicador		Descripción	
<b>Monitoreo de Usuarios</b>		Se refiere a los pasos o comandos a ejecutar para monitorear usuarios cuando se encuentran en uso de los servicios computacionales ofrecidos por la IaaS	
Forma de medición Cuantitativa según el número de pasos			
Muy Complejo	Complejo	Simple	Muy Simple
>15	>10 y <=15	>5 y <=10	<=5

Fuente: Roberto Carlos Salazar Gualoto

**Tabla III. XI Indicador Protocolo de Seguridad**

Indicador		Descripción	
<b>Protocolo de Seguridad</b>		Se refiere al uso de protocolos o credenciales de seguridad que la IaaS brinda a los usuarios para la autenticación.	
Credenciales o Protocolos de Seguridad			
Credenciales propias de Eucalyptus generadas por la Interfaz Web Protocolo X509			
Forma de medición Cuantitativa según el número de pasos			
Ninguno	Unos	Pocos	Muchos
<=0	>0 y <=2	>2 y <= 5	>5

Fuente: Roberto Carlos Salazar Gualoto

### 3.4.9. Indicadores parámetro Gestión de Nodos

**Tabla III. XII Indicador Creación, Configuración y Eliminación de Nodos**

Indicador		Descripción	
<b>Creación, Configuración y Eliminación de nodos</b>		Se refiere a los pasos que se realiza para crear configurar y eliminar nodos que son el eje principal para el despliegue de servicios	
Forma de medición Cuantitativa según el número de pasos			
Muy Complejo	Complejo	Simple	Muy Simple
>20	>15 y <=20	>10 y <= 15	<=10

Fuente: Roberto Carlos Salazar Gualoto

**Tabla III. XIII Indicador Monitoreo de Nodos**

Indicador		Descripción	
Monitoreo de Nodos		Se refiere a los pasos necesarios para monitorear nodos según el estado de despliegue del servicio	
Forma de medición Cuantitativa según el número de pasos			
Muy Complejo	Complejo	Simple	Muy Simple
>15	>10 y <=15	>5 y <=10	<=5

Fuente: Roberto Carlos Salazar Gualoto

### 3.4.10. Indicadores parámetro Gestión de Máquinas Virtuales

**Tabla III. XIV Indicador Creación, Configuración y Eliminación de Máquinas Virtuales**

Indicador		Descripción	
Creación, Configuración y Eliminación de Máquinas Virtuales		Se refiere a los pasos que se realiza para crear configurar y eliminar máquinas virtuales que son el servicio computacional que la IaaS ofrece a los usuarios.	
Forma de medición Cuantitativa según el número de pasos			
Muy Complejo	Complejo	Simple	Muy Simple
>20	>15 y <=20	>10 y <= 15	<=10

Fuente: Roberto Carlos Salazar Gualoto

**Tabla III. XV Indicador Iniciar, Reiniciar, Mostrar y Apagar Máquinas Virtuales**

Indicador		Descripción	
Iniciar, Reiniciar, Mostrar, Apagar Máquinas Virtuales		Se refiere a los pasos que se realiza para iniciar, reiniciar, mostrar, apagar máquinas virtuales	
Forma de medición Cuantitativa según el número de pasos			
Muy Complejo	Complejo	Simple	Muy Simple
>20	>15 y <=20	>10 y <= 15	<=10

Fuente: Roberto Carlos Salazar Gualoto

**Tabla III. XVI Indicador Monitoreo de Máquinas Virtuales**

Indicador		Descripción	
<b>Monitoreo de Máquinas Virtuales</b>		<b>Se refiere a los pasos necesarios para monitorear Máquinas Virtuales</b>	
Forma de medición Cuantitativa según el número de pasos			
<b>Muy Complejo</b>	<b>Complejo</b>	<b>Simple</b>	<b>Muy Simple</b>
>15	>10 y <=15	>5 y <=10	<=5

**Fuente:** Roberto Carlos Salazar Gualoto

**Tabla III. XVII Indicador Variedad de Sistemas Operativos**

Indicador		Descripción	
<b>Variedad de Sistemas Operativos</b>		<b>Se refiere al número de sistemas operativos soportados por la IaaS</b>	
Forma de medición Cuantitativa según el número de SO			
<b>Ninguno</b>	<b>Unos</b>	<b>Pocos</b>	<b>Muchos</b>
<=0	>0 y <=2	>2 y <= 5	>5

**Fuente:** Roberto Carlos Salazar Gualoto

### **3.5. Descripción de los Módulos de Prueba**

Los módulos de prueba son ambientes que ayudan a verificar y obtener datos para determinar que IaaS es el mejor a nivel de gestión y administración de servicios computacionales.

Los módulos fueron implementados en las tres IaaS seleccionadas cada una con su manera particular de gestión y provisión de servicio.

A continuación se explica cada uno de los módulos para su posterior desarrollo e implementación.

### **3.5.1. Módulo 1**

El módulo 1 fue desarrollado para probar el parámetro Configuración

En este módulo se realizó la configuración inicial de cada una de las IaaS, para evaluar comandos básicos a ejecutar, conocimientos técnicos, el tiempo y recurso humano necesario.

### **3.5.2. Módulo 2**

El módulo 2 fue desarrollado para probar el parámetro Flexibilidad

En este módulo se validó la flexibilidad que tiene la IaaS para gestionar el soporte a hipervisores, uso de DHCP, modelo de despliegue del servicio.

### **3.5.3. Módulo 3**

El módulo 3 fue desarrollado para probar el parámetro Gestión de Usuarios

El módulo describe la manera de evaluar los indicadores referentes a la creación, modificación, eliminación, monitoreo de usuarios así como también permite la identificación de credenciales de seguridad utilizadas para la autenticación de usuarios.

### **3.5.4. Módulo 4**

El módulo 4 fue desarrollado para probar el parámetro Gestión de Nodos

En este módulo se evaluó indicadores referentes a la creación configuración eliminación y monitoreo de nodos que son el eje fundamental para el despliegue de servicios computacionales ofertados por la IaaS.

### **3.5.5. Módulo 5**

El módulo 5 fue desarrollado para probar el parámetro Gestión de Máquinas Virtuales

Este módulo describe la facilidad con la que la IaaS provee servicios computacionales, permite la evaluación de los indicadores referentes a la creación configuración eliminación monitoreo de máquinas virtuales, procesos de iniciar reiniciar mostrar y apagar máquinas virtuales así como también la variedad de sistemas operativos soportados por las máquinas virtuales

## **3.6. Desarrollo de los Módulos de Prueba**

### **3.6.1. Desarrollo de Módulos de Prueba en la IaaS Eucalyptus**

#### **3.6.1.1. Módulo 1**

Para la construcción del módulo 1 se va a realizar la instalación del sistema operativo Centos 5.8 o superior, como software base y luego continuar con la instalación de la IaaS, los pasos para el desarrollo de este módulo se describen con más detalle en el Capítulo IV Implementación de la Infraestructura como Servicio Open Source Eucalyptus.

✓ **Grado de Complejidad**

Una vez realizada la configuración se determinó que para el correcto funcionamiento se requiere la ejecución de la siguiente cantidad de comandos

**Tabla III. XVIII Número de pasos a ejecutar Eucalyptus**

Fronend	Nodo	Total
45	14	59

**Fuente:** Roberto Carlos Salazar Gualoto (Capítulo IV)

✓ **Nivel de Conocimiento**

La siguiente tabla describe los índices de conocimientos técnicos necesarios para realizar la configuración de la IaaS Eucalyptus

**Tabla III. XIX Índices de Conocimiento Eucalyptus**

Índices de Conocimientos necesarios
Instalación Básica de Linux
Herramientas Básicas de Configuración
Administración y Utilización de Paquetes YUM y RPM
Creación y Eliminación de Directorios
Configuración de Gestores de Arranque
Administración Remota con SSH
Extracción de Archivos Comprimidos TAR, UNZIP, GUNZIP
Manejo de Permisos
Editores de Texto VI
Instalación y Administración de XEN
Manejo de Comandos Propios de Eucalyptus

**Fuente:** Roberto Carlos Salazar Gualoto (Capítulo IV)

✓ **Cantidad de Tiempo**

La siguiente tabla describe la cantidad de tiempo en horas de trabajo que requiere la configuración de la IaaS Eucalyptus

**Tabla III. XX Cantidad de Tiempo Eucalyptus**

Fronde End	Nodo	Total
40 horas	16 horas	56 horas

**Fuente:** Roberto Carlos Salazar Gualoto (Capítulo IV)

✓ **Cantidad de Recurso Humano**

La presente tabla describe el personal técnico necesario para la configuración de las IaaS Eucalyptus

**Tabla III. XXI Personal Técnico necesario Eucalyptus**

Fronde End	Nodo	Total
2 personas	1 persona	3 personas

**Fuente:** Roberto Carlos Salazar Gualoto (Capítulo IV)

**3.6.1.2. Módulo 2**

✓ **Cantidad de Hypervisores**

Revisada información referente al soporte de hypervisores se encontró que la IaaS Eucalyptus posee soporte a los hypervisores detallados a continuación.

**Tabla III. XXII Soporte Hypervisores Eucalyptus**

Soporte Hypervisores
XEN
KVM
VMware

**Fuente:** Roberto Carlos Salazar Gualoto (Capítulo IV)

✓ **Grado de Uso de DHCP**

En las pruebas realizadas se determinó que la IaaS Eucalyptus presenta el uso de DHCP de manera dinámica no requiere la configuración dentro de IaaS de este servicio.

✓ **Nivel de Despliegue**

Investigada información concerniente al despliegue de servicios se encontró que el mismo es realizado de manera dinámica que no requiere ejecución extra de comandos.

✓ **Cantidad de Entornos de Programación**

Analizando información referente a entornos de programación mediante los cuales ha sido desarrollada la IaaS e instalado componentes para los mismos, se encontró los siguientes entornos listados en la siguiente tabla.

**Tabla III. XXIII Entornos de Programación Eucalyptus**

Entornos de Programación
Hibernate
Axis
Java

**Fuente:** Roberto Carlos Salazar Gualoto (Capítulo IV)

**3.6.1.3. Módulo 3**

En este módulo se prueba las funcionalidades referentes a la gestión de usuarios que se explica detalladamente en el Capítulo IV

✓ **Creación, Modificación, Eliminación de Usuarios**

Ejecutados comandos referentes a este indicador se ha cuantificado lo que se detalla en la siguiente tabla

**Tabla III. XXIV Número de Comandos para Creación, Modificación, Eliminación de Usuarios Eucalyptus**

Creación	Modificación	Eliminación	Total
6	5	2	13

**Fuente:** Roberto Carlos Salazar Gualoto (Capítulo IV)

✓ **Monitoreo de Usuarios**

Ejecutados comandos referentes a este indicador se ha cuantificado lo que se detalla en la siguiente tabla

**Tabla III. XXV Número de Comandos Monitoreo Usuarios Eucalyptus**

Monitoreo	Total
3	3

**Fuente:** Roberto Carlos Salazar Gualoto (Capítulo IV)

✓ **Protocolo de Seguridad**

Descargado los componentes necesarios para la autenticación de usuarios se ha encontrado que la interfaz WEB de Eucalyptus proporciona Credenciales propias de Seguridad basadas en el protocolo de Seguridad X509.

**3.6.1.4. Módulo 4**

✓ **Creación, Configuración y Eliminación de Nodos**

Una vez realizada la configuración del nodo de Eucalyptus descrita a mayor detalle en el Capítulo IV se procede a cuantificar el número de comandos ejecutados para la realización de este indicador detallándose a continuación en la siguiente Tabla.

**Tabla III. XXVI Número de Comandos para Creación, Configuración, Eliminación de Nodos Eucalyptus**

Creación	Configuración	Eliminación	Total
2	2	1	5

**Fuente:** Roberto Carlos Salazar Gualoto (Capítulo IV)

✓ **Monitoreo de Nodos**

Realizada la configuración se estableció que los comandos ejecutados para la validación de este indicador son detallados en la siguiente tabla.

**Tabla III. XXVII Número de Comandos Monitoreo de Nodos Eucalyptus**

Monitoreo	Total
4	4

**Fuente:** Roberto Carlos Salazar Gualoto (Capítulo IV)

**3.6.1.5. Módulo 5**

✓ **Creación, Configuración Eliminación de Máquinas Virtuales**

En las pruebas realizadas se pudo cuantificar los comandos ejecutados que se detalla siguiente tabla.

**Tabla III. XXVIII Número de Comandos para Creación, Configuración, Eliminación de Máquinas Virtuales Eucalyptus**

Creación	Configuración	Eliminación	Total
4	4	2	10

**Fuente:** Roberto Carlos Salazar Gualoto (Capítulo IV)

✓ **Iniciar, Reiniciar Mostrar y Apagar de Máquinas Virtuales**

Realizada pruebas para valorar este indicador se pudo cuantificar los comandos ejecutados que se detalla siguiente tabla.

**Tabla III. XXIX Número de Comandos para Iniciar, Reiniciar, Mostrar y Apagar Máquinas Virtuales Eucalyptus**

Iniciar	Reiniciar	Mostrar	Apagar	Total
2	1	1	1	5

**Fuente:** Roberto Carlos Salazar Gualoto

✓ **Monitoreo de Máquinas Virtuales**

Para la validación de este indicador pudimos cuantificar la ejecución de comandos que se detalla siguiente tabla.

**Tabla III. XXX Número de Comandos para Monitoreo de Máquinas Virtuales Eucalyptus**

Monitoreo	Total
3	3

**Fuente:** Roberto Carlos Salazar Gualoto (Capítulo IV)

✓ **Variedad de Sistemas Operativos**

Revisado las imágenes certificadas de Eucalyptus se determinó que la IaaS presenta soporte a varias distribuciones del SO Linux y creando imágenes también soporta SO Windows.

**3.6.2. Desarrollo de Módulos de Prueba en la IaaS Nimbus**

**3.6.2.1. Módulo 1**

Para la construcción del módulo 1 se va a realizar la instalación del sistema operativo Centos 5.6 o superior, como software base y luego continuar con la instalación de la IaaS, los pasos para el desarrollo de este módulo se describen con mayor detalle en el Anexo1 Instalación y Administración de Nimbus.

✓ **Grado de Complejidad**

Una vez realizada la configuración se determinó que para el correcto funcionamiento se requiere la ejecución de la siguiente cantidad de comandos.

**Tabla III. XXXI Número de pasos a ejecutar Nimbus**

Fronde End	Nodo	Total
77	35	132

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 1)

✓ **Nivel de Conocimiento**

La siguiente tabla describe los índices de conocimientos técnicos necesarios para realizar la configuración de la IaaS Nimbus

**Tabla III. XXXII Índices de Conocimiento Nimbus**

Índices de Conocimientos necesarios
Instalación Básica de Linux
Herramientas Básicas de Configuración
Administración y Utilización de Paquetes YUM y RPM
Creación y Eliminación de Directorios
Configuración de Gestores de Arranque
Administración Remota con SSH
Extracción de Archivos Comprimidos TAR, UNZIP, GUNZIP
Manejo de Permisos
Editores de Texto VI
Gestión de Usuarios y Contraseñas
Herramienta de descarga de contenidos WGET
Instalación de Python
Servicio DHCP
Configuración de Bridge
Instalación y Administración de XEN
Manejo de Comandos Propios de Nimbus

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 1)

✓ **Cantidad de Tiempo**

La siguiente tabla describe la cantidad de tiempo en horas de trabajo que requiere la configuración de la IaaS Nimbus

**Tabla III. XXXIII Cantidad de Tiempo Nimbus**

Fronde End	Nodo	Total
40 horas	40 horas	80 horas

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 1)

✓ **Cantidad de Recurso Humano**

La presente tabla describe el personal técnico necesario para la configuración de las IaaS Nimbus

**Tabla III. XXXIV Personal Técnico necesario Nimbus**

Fronde End	Nodo	Total
2 personas	2 personas	4 personas

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 1)

**3.6.2.2. Módulo 2**

✓ **Cantidad de Hypervisores**

Analizada información referente al soporte de hypervisores se encontró que la IaaS Nimbus posee soporte a los hypervisores detallados a continuación.

**Tabla III. XXXV Soporte Hypervisores Nimbus**

Soporte Hypervisores
XEN
KVM

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 1)

✓ **Grado de Uso de DHCP**

En las pruebas realizadas se determinó que la IaaS Nimbus presenta el uso de DHCP de manera automática, requiere la configuración dentro de IaaS de este servicio.

✓ **Nivel de Despliegue**

Investigada información concerniente al despliegue de servicios se encontró que el mismo es realizado de manera dinámica que no requiere ejecución extra de comandos.

✓ **Cantidad de Entornos de Programación**

Analizada información referente a entornos de programación mediante los cuales han sido desarrollada la IaaS e instalado componentes para los mismos, se encontró los siguientes entornos listados en la siguiente tabla.

**Tabla III. XXXVI Entornos de Programación Eucalyptus**

Entornos de Programación
Python
Java

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 1)

**3.6.2.3. Módulo 3**

En este módulo se prueba las funcionalidades referentes a la gestión de usuarios que se explica detalladamente en el Anexo 1.

✓ **Creación, Modificación, Eliminación de Usuarios**

Ejecutados comandos referentes a este indicador se ha cuantificado lo que se detalla en la siguiente tabla

**Tabla III. XXXVII Número de Comandos para Creación, Modificación, Eliminación de Usuarios Nimbus**

Creación	Modificación	Eliminación	Total
8	4	4	16

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 1)

✓ **Monitoreo de Usuarios**

Ejecutados comandos referentes a este indicador se ha cuantificado lo que se detalla en la siguiente tabla

**Tabla III. XXXVIII Número de Comandos Monitoreo Usuarios Nimbus**

Monitoreo	Total
3	3

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 1)

✓ **Protocolo de Seguridad**

Descargado los componentes necesarios para la autenticación de usuarios se encontró que la IaaS Nimbus usa el protocolo de Seguridad X509 como credenciales de seguridad.

**3.6.2.4. Módulo 4**

✓ **Creación, Configuración y Eliminación de Nodos**

Una vez realizada la configuración del nodo de Nimbus descrita a mayor detalle en el Anexo 1 se procede a cuantificar el número de comandos ejecutados para la realización de este indicador detallándose a continuación en la siguiente Tabla.

**Tabla III. XXXIX Número de Comandos para Creación, Configuración, Eliminación de Nodos Nimbus**

Creación	Configuración	Eliminación	Total
4	9	3	16

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 1)

✓ **Monitoreo de Nodos**

Realizada la configuración se cuantificó los comandos ejecutados para la validación de este indicador, detallado en la siguiente tabla.

**Tabla III. XXXX Número de Comandos Monitoreo de Nodos Nimbus**

Monitoreo	Total
4	4

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 1)

**3.6.2.5. Módulo 5**

✓ **Creación, Configuración Eliminación de Máquinas Virtuales**

En las pruebas realizadas se pudo cuantificar los comandos ejecutados que se detalla siguiente tabla.

**Tabla III. XXXXI Número de Comandos para Creación, Configuración, Eliminación de Máquinas Virtuales Nimbus**

Creación	Configuración	Eliminación	Total
3	3	1	7

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 1)

✓ **Iniciar, Reiniciar, Mostrar y Apagar de Máquinas Virtuales**

Realizada pruebas para valorar este indicador se pudo cuantificar los comandos ejecutados que se detalla siguiente tabla.

**Tabla III. XXXXII Número de Comandos para Iniciar, Reiniciar Mostrar y Apagar Máquinas Virtuales Nimbus**

Iniciar	Reiniciar	Mostrar	Apagar	Total
1	1	1	1	4

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 1)

✓ **Monitoreo de Máquinas Virtuales**

Para la validación de este indicador pudimos cuantificar la ejecución de comandos que se detalla siguiente tabla.

**Tabla III. XXXXIII Número de Comandos para Monitoreo de Máquinas Virtuales**

**Nimbus**

Monitoreo	Total
3	3

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 1)

✓ **Variedad de Sistemas Operativos**

Revisando información referente a soporte a Sistemas Operativos que la IaaS presenta se encontró que soporta a varias distribuciones del Sistema Operativo Linux

**3.6.3. Desarrollo de Módulos de Prueba en la IaaS OpenNebula**

**3.6.3.1. Módulo 1**

Para la construcción del módulo 1 se va a realizar la instalación del sistema operativo Centos 5.6 o superior, como software base y luego continuar con la instalación de la IaaS, los pasos para el desarrollo de este módulo se describen con mayor detalle en el Anexo 2 Instalación y Administración de OpenNebula.

✓ **Grado de Complejidad**

Una vez realizada la configuración se determinó que para el correcto funcionamiento se requiere la ejecución de la siguiente cantidad de comandos

**Tabla III. XXXXIV Número de pasos a ejecutar OpenNebula**

Fronde End	Nodo	Total
135	26	161

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 2)

✓ **Nivel de Conocimiento**

La siguiente tabla describe los índices de conocimientos técnicos necesarios para realizar la configuración de la IaaS OpenNebula

**Tabla III. XXXXV Índices de Conocimiento OpenNebula**

Índices de Conocimientos necesarios
Instalación Básica de Linux
Herramientas Básicas de Configuración
Administración y Utilización de Paquetes YUM y RPM
Creación y Eliminación de Directorios
Configuración de Gestores de Arranque
Administración Remota con SSH
Extracción de Archivos Comprimidos TAR, UNZIP, GUNZIP
Manejo de Permisos
Editores de Texto VI
Gestión de Usuarios y Contraseñas
Herramienta de descarga de contenidos WGET, GEM
Instalación de Ruby
Instalación y Administración de XEN
Delegación de Tareas Superusuario sudo
Manejo de Comandos Propios de OpenNebula

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 2)

✓ **Cantidad de Tiempo**

La siguiente tabla describe la cantidad de tiempo en horas de trabajo que requiere la configuración de la IaaS OpenNebula

**Tabla III. XXXXVI Cantidad de Tiempo OpenNebula**

Fronde End	Nodo	Total
48 horas	48 horas	96 horas

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 2)

✓ **Cantidad de Recurso Humano**

La presente tabla describe el personal técnico necesario para la configuración de las IaaS OpenNebula.

**Tabla III. XXXXVII Personal Técnico necesario OpenNebula**

Front End	Nodo	Total
2 personas	2 personas	4 personas

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 2)

**3.6.3.2. Módulo 2**

✓ **Cantidad de Hypervisores**

Revisada información referente al soporte de hypervisores se encontró que la IaaS OpenNebula posee soporte a los siguientes hypervisores detallados a continuación.

**Tabla III. XXXXVIII Soporte Hypervisores Eucalyptus**

Soporte Hypervisores
XEN
KVM

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 2)

✓ **Grado de Uso de DHCP**

En las pruebas realizadas se determinó que la IaaS OpenNebula presenta el uso de DHCP de manera estática es decir se configura un archivo de red que determina la dirección ip a la máquina virtual brindada como servicio.

✓ **Nivel de Despliegue**

Investigada información concerniente al despliegue de servicios se encontró que el mismo es realizado de manera dinámica que no requiere ejecución extra de comandos.

✓ **Cantidad de Entornos de Programación**

Analizada información referente a entornos de programación mediante los cuales ha sido desarrollada la IaaS e instalado componentes para los mismos, se encontró los siguientes entornos listados en la siguiente tabla.

**Tabla III. XXXXIX Entornos de Programación OpenNebula**

Entornos de Programación
Ruby
Java

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 2)

**3.6.3.3. Módulo 3**

En este módulo se prueba las funcionalidades referentes a la gestión de usuarios que se explica detalladamente en el Anexo 2

✓ **Creación, Modificación, Eliminación de Usuarios**

Ejecutados comandos referentes a este indicador se ha cuantificado lo que se detalla en la siguiente tabla

**Tabla III. XXXXX Número de Comandos para Creación, Modificación, Eliminación de Usuarios OpenNebula**

Creación	Modificación	Eliminación	Total
4	2	1	7

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 2)

✓ **Monitoreo de Usuarios**

Ejecutados comandos referentes a este indicador se ha cuantificado lo que se detalla en la siguiente tabla

**Tabla III. XXXXXI Número de Comandos Monitoreo OpenNebula**

Monitoreo	Total
3	3

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 2)

✓ **Protocolo de Seguridad**

Descargado los componentes necesarios para la autenticación de usuarios se ha encontrado que la IaaS OpenNebula usa el protocolo de Seguridad X509 como credencial de seguridad.

**3.6.3.4. Módulo 4**

✓ **Creación, Configuración y Eliminación de Nodos**

Una vez realizada la configuración del nodo de OpenNebula descrita a mayor detalle en el Anexo 2 se procede a cuantificar el número de comandos ejecutados para la realización de este indicador detallándose a continuación en la siguiente tabla.

**Tabla III. XXXXXII Número de Comandos para Creación, Configuración, Eliminación de Nodos OpenNebula**

Creación	Configuración	Eliminación	Total
4	4	1	9

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 2)

✓ **Monitoreo de Nodos**

Realizada la configuración se estableció que los comandos ejecutados para la validación de este indicador son detallados en la siguiente tabla.

**Tabla III. XXXXXIII Número de Comandos Monitoreo de Nodos OpenNebula**

Monitoreo	Total
4	4

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 2)

### 3.6.3.5. Módulo 5

#### ✓ **Creación, Configuración Eliminación de Máquinas Virtuales**

En las pruebas realizadas se pudo cuantificar los comandos ejecutados que se detalla siguiente tabla.

**Tabla III. XXXXXIV Número de Comandos para Creación, Configuración, Eliminación de Máquinas Virtuales OpenNebula**

Creación	Configuración	Eliminación	Total
8	7	2	17

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 2)

#### ✓ **Iniciar, Reiniciar Mostrar y Apagar de Máquinas Virtuales**

Realizada pruebas para valorar este indicador se pudo cuantificar los comandos ejecutados se detallan en la siguiente tabla.

**Tabla III. XXXXXV Número de Comandos para Iniciar, Reiniciar Mostrar y Apagar Máquinas Virtuales OpenNebula**

Iniciar	Reiniciar	Mostrar	Apagar	Total
4	2	2	2	10

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 2)

✓ **Monitoreo de Máquinas Virtuales**

Para la validación de este indicador pudimos cuantificar la ejecución de comandos que se detalla siguiente tabla.

**Tabla III. XXXXXVI Número de Comandos para Monitoreo de Máquinas Virtuales**  
**OpenNebula**

Monitoreo	Total
3	3

**Fuente:** Roberto Carlos Salazar Gualoto (Anexo 2)

✓ **Variedad de Sistemas Operativos**

Revisado las imágenes que soporta OpenNebula se determinó que la IaaS presenta soporte a varias distribuciones del Sistema Operativo Linux.

### **3.7. Análisis Comparativo**

En esta sección se va a mostrar el análisis de tecnologías de Infraestructuras como Servicios (IaaS), Eucalyptus, Nimbus y OpenNebula a manera de cuadros comparativos, seguidos de la interpretación y la calificación del indicador evaluado por parte del autor, estos cuadros comparativos se encuentran clasificados de acuerdo a los parámetros de comparación definidos anteriormente.

Para alcanzar los resultados cuantitativos y cualitativos que permitan una selección sustentada de una de las tecnologías de Infraestructura como Servicio analizadas, se calificó cada uno de los parámetros de comparación, basados en la siguiente escala:

**Tabla III. XXXXXVII Escala Cualitativa para los Indicadores**

Regular	Bueno	Muy Bueno	Excelente
<70%	>=70% y <80%	>=80% y <90%	>=90%

**Fuente:** Roberto Carlos Salazar Gualoto

**Tabla III. XXXXXVIII Escala Cuantitativa para los Indicadores**

CUANTITATIVA			
1	2	3	4
Muy Bajo	Bajo	Medio	Alto
Deficiente	Poco Eficiente	Eficiente	Muy Eficiente
Mala	Regular	Buena	Muy Buena
Muy Complejo	Complejo	Simple	Muy Simple
Ninguno	Parcialmente	En su Mayor Parte	Totalmente
Ninguno	Uno	Pocos	Muchos

**Fuente:** Roberto Carlos Salazar Gualoto

Cada uno de los ítems de la interpretación incluye la siguiente nomenclatura:

(w,x,y)/z en donde cada letra significa lo siguiente:

w: representa el puntaje que obtiene Eucalyptus

x: representa el puntaje que obtiene Nimbus

y: representa el puntaje que obtiene OpenNebula

z: representa la base del puntaje sobre la cual está evaluado el parámetro

La calificación definitiva de la tecnología de Infraestructura como Servicio en base a cada parámetro de comparación se obtiene sumando los puntajes obtenidos del análisis, utilizando las siguientes fórmulas:

$$Peus = \sum(w), Pnis = \sum(x), Popa = \sum(y), Pc = \sum(z)$$

$$\text{Calificación de Eucalyptus (Cc - Eus)} = (Peus / Pc) * 100\%$$

$$\text{Calificación de Nimbus (Cc - Nis)} = (Pnis / Pc) * 100\%$$

$$\text{Calificación de OpenNebula (Cc - Opa)} = (Popa / Pc) * 100\%$$

**En donde:**

Peus: puntaje acumulado por Eucalyptus en el parámetro

Pnis: puntaje acumulado por Nimbus en el parámetro

Popa: puntaje acumulado por OpenNebula en el parámetro

Pc: puntaje sobre el cual se califica al parámetro

Cc – Eus: porcentaje de la calificación total que obtuvo Eucalyptus en el parámetro.

Cc – Nis: porcentaje de la calificación total que obtuvo Nimbus en el parámetro.

Cc – Opa: porcentaje de la calificación total que obtuvo OpenNebula en el parámetro.

### **3.7.1. Parámetro Configuración**

#### **3.7.1.1. Determinación de Variables**

- a) Grado de Complejidad
- b) Nivel de Conocimiento
- c) Cantidad de Tiempo
- d) Cantidad de Recurso Humano

#### **3.7.1.2. Valoraciones**

**a) Grado de Complejidad**

Valoración: 4 puntos

**b) Nivel de Conocimiento**

Valoración: 4 puntos

**c) Cantidad de Tiempo**

Valoración: 4 puntos

**d) Cantidad de Recurso Humano**

Valoración: 4 puntos

**Tabla III. XXXXXIX Valoración Parámetro Configuración**

Indicador	Eucalyptus	Nimbus	OpenNebula
Grado de Complejidad	Bajo	Medio	Alto
Nivel de Conocimiento	Medio	Alto	Alto
Cantidad de Tiempo	Pocos	Pocos	Muchos
Cantidad de Recurso Humano	Pocos	Pocos	Pocos

**Fuente:** Roberto Carlos Salazar Gualoto

**3.7.1.3. Interpretación**

- ✓ Eucalyptus requiere 59 pasos, Nimbus requiere 132 pasos y OpenNebula 161 pasos para su implementación inicial, estos resultados han sido cuantificados en base a la Tabla III. II Indicador Grado de Complejidad. (2,3,4)/4
- ✓ Se encontró que Eucalyptus requiere 11 índices de conocimientos técnicos, Nimbus 15, OpenNebula 16 por lo que se requiere un nivel alto de

conocimientos, este resultado ha sido valorado en base a la Tabla III. III  
Indicador Nivel de Conocimiento (3,4,4)/4

- ✓ La cantidad de tiempo requerida en Eucalyptus son 56 horas de trabajo, Nimbus 80 y OpenNebula 120, por lo que la cantidad de tiempo necesario para la implementación difieren en Pocos y Muchos días respectivamente según la Tabla III. IV Indicador Cantidad de Tiempo en la cual se basa el resultado obtenido. (3,3,4)/4
- ✓ El recurso humano necesario para la implementación es, Eucalyptus 3 personas, Nimbus 4 y OpenNebula 4 personas lo que nos indica que la cantidad de recurso humanos necesarios en las IaaS comparadas son Pocos resultado obtenido en base a la Tabla III. V Indicador Cantidad de Recurso Humano. (3,3,3)/4

#### 3.7.1.4. Calificación

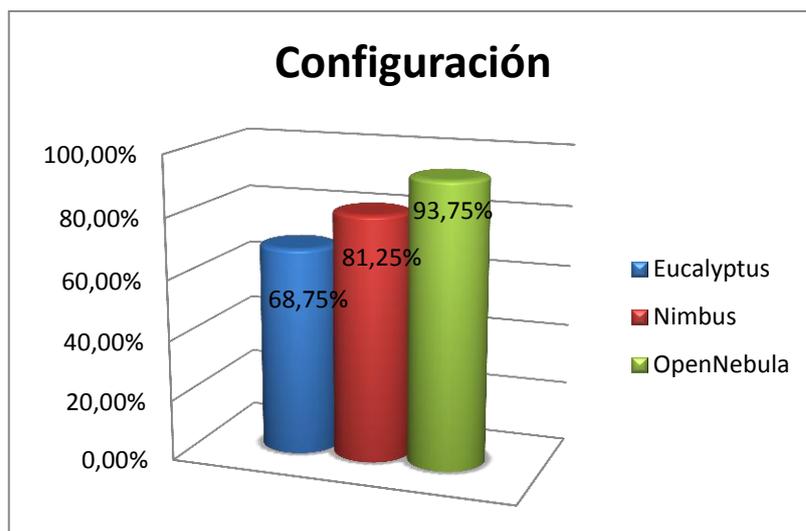


Figura III - 16. Comparación Porcentajes Parámetro Configuración

$$Pc = \sum(z) = 4 + 4 + 4 + 4 = 16$$

$$Peus = \sum(x) = 2 + 3 + 3 + 3 = 11$$

$$Pnis = \sum(x) = 3 + 4 + 3 + 3 = 13$$

$$Popa = \sum(x) = 4 + 4 + 4 + 3 = 15$$

$$Cc - Eus = \frac{Peus}{Pc} = \left(\frac{11}{16}\right) * 100\% = 68,75\%$$

$$Cc - Nis = \frac{Pnis}{Pc} = \left(\frac{13}{16}\right) * 100\% = 81,25\%$$

$$Cc - Opa = \frac{Popa}{Pc} = \left(\frac{15}{16}\right) * 100\% = 93,75\%$$

### 3.7.2. Parámetro Flexibilidad

#### 3.7.2.1. Determinación de Variables

- a) Cantidad de Soporte Hypervisor
- b) Grado de Uso de DHCP
- c) Nivel de Despliegue
- d) Cantidad de Entornos de Programación

#### 3.7.2.2. Valoraciones

##### a) Cantidad de Soporte Hypervisor

Valoración: 4 puntos

##### b) Grado de Uso de DHCP

Valoración: 4 puntos

##### c) Nivel de Despliegue

Valoración: 4 puntos

##### d) Cantidad de Entorno de Programación

Valoración: 4 puntos

**Tabla III. XXXXXX Valoración Parámetro Flexibilidad**

Indicador	Eucalyptus	Nimbus	OpenNebula
Cantidad Soporte Hypervisor	Pocos	Unos	Unos
Grado de Uso de DHCP	Alto	Medio	Bajo
Nivel de Despliegue	Alto	Alto	Alto
Cantidad de Entornos de Programación	Pocos	Unos	Unos

**Fuente:** Roberto Carlos Salazar Gualoto

### 3.7.2.3. Interpretación

- ✓ Eucalyptus presenta soporte a tres Hypervisores XEN, KVM y VMware, Nimbus y OpenNebula dos XEN y KVM estos resultados han sido valorados en base a la Tabla III. V Indicador Cantidad Soporte Hypervisor. (3,2,2)/4
- ✓ Eucalyptus requiere la existencia de DHCP de manera dinámica, Nimbus de manera automática y OpenNebula de manera estática resultados evaluados en base a la Tabla III. VI Indicador Grado de Uso de DHCP. (4,3,2)/4
- ✓ La tres IaaS usan el nivel de despliegue de manera dinámica resultado justificado en base a la Tabla III. VII Indicador Nivel de Despliegue. (4,4,4)/4
- ✓ Eucalyptus soporta varios entornos Hibernate, Axis y Java, Nimbus Python y Java y OpenNebula Ruby y Java datos evaluados en base a la Tabla III. VIII Indicador Entornos de Programación. (3,2,2)/4

### 3.7.2.4. Calificación

$$Pc = \sum(z) = 4 + 4 + 4 + 4 = 16$$

$$Peus = \sum(x) = 3 + 4 + 4 + 3 = 14 \quad Cc - Eus = \frac{Peus}{Pc} = \left(\frac{14}{16}\right) * 100\% = 87,50\%$$

$$Pnis = \sum(x) = 2 + 3 + 4 + 2 = 11 \quad Cc - Nis = \frac{Pnis}{Pc} = \left(\frac{11}{16}\right) * 100\% = 68,75\%$$

$$Popa = \sum(x) = 2 + 2 + 4 + 2 = 10 \quad Cc - Opa = \frac{Popa}{Pc} = \left(\frac{10}{16}\right) * 100\% = 62,50\%$$

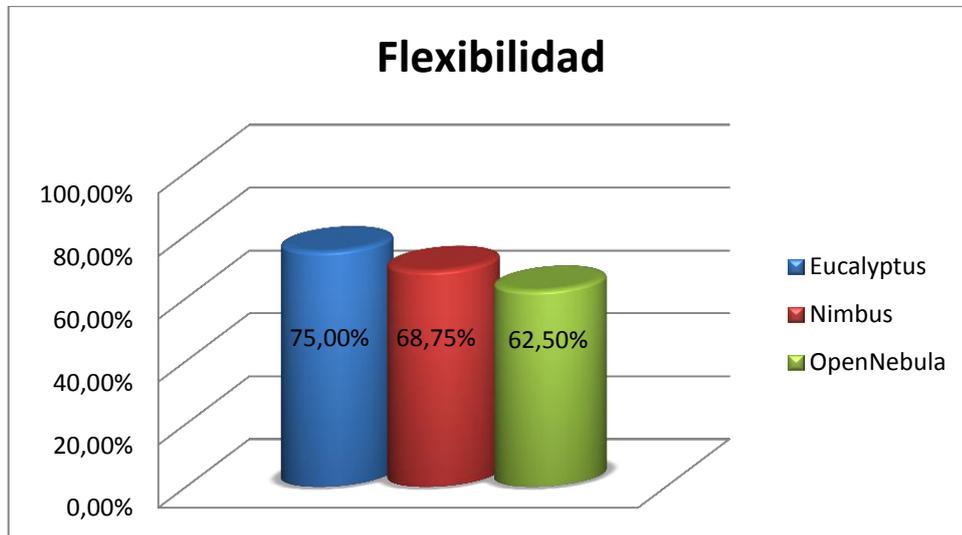


Figura III - 17. Comparación Porcentajes Parámetro Flexibilidad

### 3.7.3. Parámetro Gestión de Usuarios

#### 3.7.3.1. Determinación de Variables

- a) Creación, Modificación, Eliminación de Usuarios
- b) Monitoreo de Usuarios
- c) Protocolo de Seguridad

### 3.7.3.2. Valoraciones

#### a) Creación, Modificación, Eliminación de Usuarios

Valoración: 4 puntos

#### b) Monitoreo de Usuarios

Valoración: 4 puntos

#### c) Protocolo de Seguridad

Valoración: 4 puntos

**Tabla III. XXXXXXI Valoración Parámetro Gestión de Usuarios**

Indicador	Eucalyptus	Nimbus	OpenNebula
Creación, Modificación, Eliminación de Usuarios	Simple	Complejo	Muy Simple
Monitoreo de Usuarios	Muy Simple	Muy Simple	Muy Simple
Protocolo de Seguridad	Unos	Unos	Unos

**Fuente:** Roberto Carlos Salazar Gualoto

### 3.7.3.3. Interpretación

- ✓ La IaaS Eucalyptus para la gestión de usuarios requiere la ejecución de 13 pasos en el entorno WEB, Nimbus en el entorno CLI requiere una mayor cantidad de pasos 16 y OpenNebula requiere una cantidad menor de pasos 7 para la creación modificación y eliminación de usuarios datos valorados en base a la Tabla III. IX Indicador Creación, Modificación, Eliminación de Usuarios. (3,2,4)/4
- ✓ Las tres IaaS ya sea por el entorno WEB o CLI presentan una mínima cantidad de pasos para el monitoreo de usuarios datos que han sido evaluados en base a la Tabla III. X Indicador Monitoreo de Usuarios. (4,4,4)/4

- ✓ Eucalyptus usa credenciales de seguridad propias de la tecnología que provee su interfaz web. Nimbus credenciales Publicas X509 al igual que OpenNebula datos justificados en base a la Tabla III. XI Indicador Protocolos de Seguridad. (2,2,2)/4

### 3.7.3.4. Calificación

$$Pc = \sum(z) = 4 + 4 + 4 = 12$$

$$Peus = \sum(x) = 3 + 4 + 2 = 9 \quad Cc - Eus = \frac{Peus}{Pc} = \left(\frac{9}{12}\right) * 100\% = 75\%$$

$$Pnis = \sum(x) = 2 + 4 + 2 = 8 \quad Cc - Nis = \frac{Pnis}{Pc} = \left(\frac{8}{12}\right) * 100\% = 66,67\%$$

$$Popa = \sum(x) = 4 + 4 + 2 = 10 \quad Cc - Opa = \frac{Popa}{Pc} = \left(\frac{10}{12}\right) * 100\% = 83,34\%$$



Figura III - 18. Comparación Porcentajes Parámetro Gestión de Usuarios

### 3.7.4. Parámetro Gestión de Nodos

#### 3.7.4.1. Determinación de Variables

- a) Creación, Configuración y Eliminación de Nodos
- b) Monitoreo de Nodos:

#### 3.7.4.2. Valoraciones

##### a) Creación, Configuración y Eliminación de Nodos

Valoración: 4 puntos

##### b) Monitoreo de Nodos

Valoración: 4 puntos

**Tabla III. XXXXXXII Valoración Parámetro Gestión de Nodos**

Indicador	Eucalyptus	Nimbus	OpenNebula
Creación, Configuración y Eliminación de Nodos	Muy Simple	Complejo	Muy Simple
Monitoreo de Nodos	Muy Simple	Muy Simple	Muy Simple

**Fuente:** Roberto Carlos Salazar Gualoto

#### 3.7.4.3. Interpretación

- ✓ Eucalyptus y OpenNebula en pocos pasos presentan las opciones de creación configuración y eliminación de nodos, mientras que Nimbus utiliza una mayor cantidad de pasos resultados basados en la Tabla III. XII Indicador Creación, Configuración y Eliminación de Nodos.(4,2,4)/4

- ✓ El monitoreo de Nodos en las tres IaaS se realiza en pocos pasos por lo que las IaaS difieren en la forma propia de realizar dicha operación resultados en base a la Tabla III. XIII Indicador Monitoreo de Nodos. (4,4,4)/4

#### 3.7.4.4. Calificación

$$Pc = \sum(z) = 4 + 4 = 8$$

$$Peus = \sum(x) = 4 + 4 = 8 \quad Cc - Eus = \frac{Peus}{Pc} = \left(\frac{8}{8}\right) * 100\% = 100\%$$

$$Pnis = \sum(x) = 2 + 4 = 6 \quad Cc - Nis = \frac{Pnis}{Pc} = \left(\frac{6}{8}\right) * 100\% = 75\%$$

$$Popa = \sum(x) = 4 + 4 = 8 \quad Cc - Opa = \frac{Popa}{Pc} = \left(\frac{8}{8}\right) * 100\% = 100\%$$

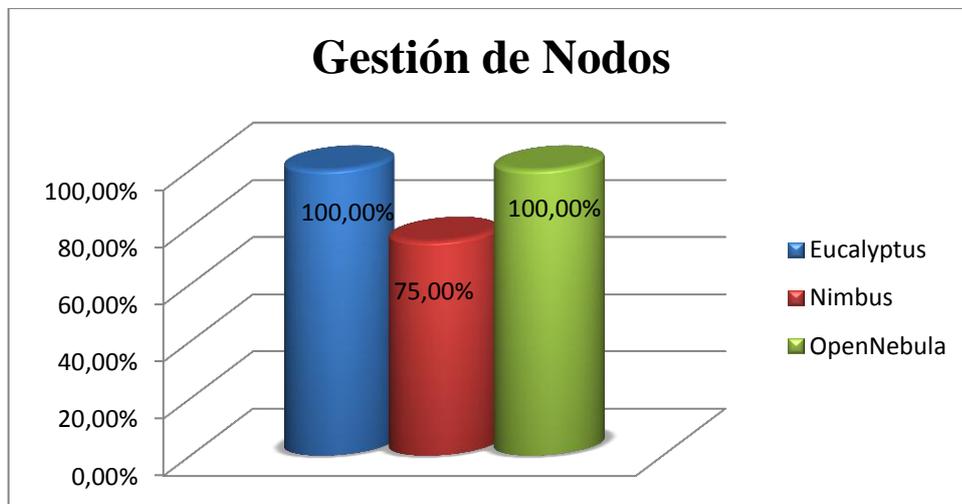


Figura III - 19. Comparación Porcentajes Parámetro Gestión de Nodos

#### 3.7.5. Parámetro Gestión de Máquinas Virtuales

##### 3.7.5.1. Determinación de Variables

- a) Creación, Configuración y Eliminación de Máquinas Virtuales

- b) Iniciar, Reiniciar, Mostrar y Apagar Máquinas Virtuales
- c) Monitoreo de Máquinas Virtuales
- d) Variedad de Sistemas Operativos

### 3.7.5.2. Valoraciones

✓ **Creación, Configuración y Eliminación de Máquinas Virtuales**

Valoración: 4 puntos

✓ **Iniciar, Reiniciar, Mostrar y Apagar Máquinas Virtuales**

Valoración: 4 puntos

✓ **Monitoreo de Máquinas Virtuales**

Valoración: 4 puntos

✓ **Variedad de Sistemas Operativos**

Valoración: 4 puntos

**Tabla III. XXXXXXIII Valoración Parámetro Gestión de Máquinas Virtuales**

Indicador	Eucalyptus	Nimbus	OpenNebula
Creación, Configuración y Eliminación de Máquinas Virtuales	Muy Simple	Muy Simple	Complejo
Iniciar, Reiniciar, Mostrar y Apagar Máquinas Virtuales	Muy Simple	Muy Simple	Muy Simple
Monitoreo de Máquinas Virtuales	Muy Simple	Muy Simple	Muy Simple
Variedad de Sistemas Operativos	Unos	Unos	Unos

**Fuente:** Roberto Carlos Salazar Gualoto

### 3.7.5.3. Interpretación

- ✓ Eucalyptus y Nimbus requieren la ejecución de un número menor de comandos propios de cada IaaS para el despliegue de servicio siendo esto fundamental la

creación, configuración y eliminación de máquinas virtuales para el otorgamiento de servicios a los clientes, OpenNebula requiere mayor número de pasos, resultados evaluados en base a la Tabla III. XIV Indicador Creación, Configuración y Eliminación de Máquinas Virtuales.(4,4,2)/4

- ✓ Las tres IaaS requieren un número menor de comandos a ejecutar para iniciar, reiniciar, mostrar y apagar máquinas virtuales entregadas como servicio por las Infraestructuras, resultados justificados en base a la Tabla III. XV Indicador Iniciar, Reiniciar, Mostrar y Apagar Máquinas Virtuales. (4,4,4)/4
  
- ✓ Las IaaS analizadas requieren un número pequeño de comandos a ejecutar para realizar el monitoreo de servicio a entregar en este caso máquinas virtuales, resultados basados en la Tabla III. VI Indicador Monitoreo de Máquinas Virtuales. (4,4,4)/4
  
- ✓ Eucalyptus, Nimbus y OpenNebula poseen soporte de sistemas operativos LINUX es decir en las tres IaaS se puede correr distribuciones de SO LINUX mientras que en Eucalyptus creando una imagen se puede crear SO Windows, resultados evaluados en base a la Tabla III. VII Indicador Variedad de Sistemas Operativos. (2,2,2)/4

#### **3.7.5.4. Calificación**

$$Pc = \sum(z) = 4 + 4 + 4 + 4 = 16$$

$$Peus = \sum(x) = 4 + 4 + 4 + 2 = 14 \quad Cc - Eus = \frac{Peus}{Pc} = \left(\frac{14}{16}\right) * 100\% = 87,50\%$$

$$Pnis = \sum(x) = 4 + 4 + 4 + 2 = 14 \quad Cc - Nis = \frac{Pnis}{Pc} = \left(\frac{14}{16}\right) * 100\% = 87,50\%$$

$$Popa = \sum(x) = 2 + 4 + 4 + 2 = 12 \quad Cc - Opa = \frac{Popa}{Pc} = \left(\frac{12}{16}\right) * 100\% = 75\%$$

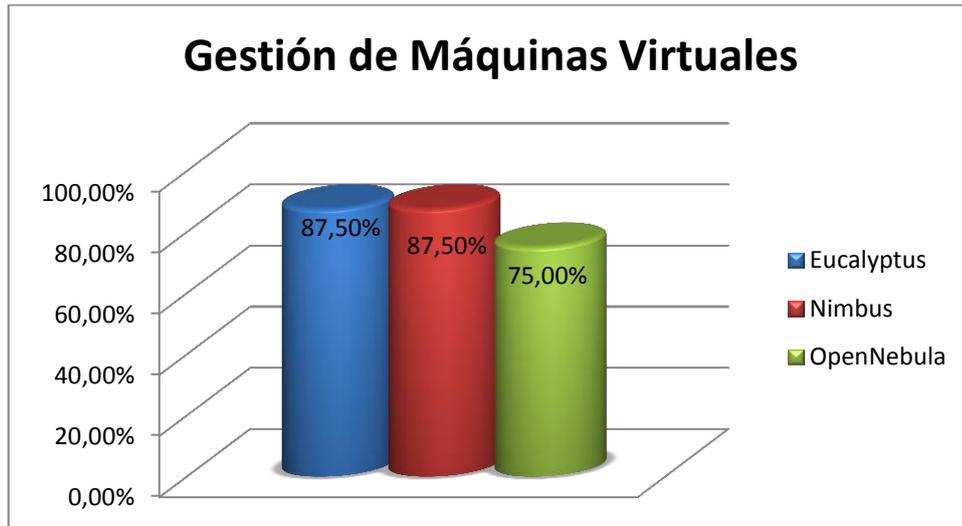


Figura III - 20. Comparación Porcentajes Parámetro Gestión de Máquinas Virtuales

### 3.7.6. Puntajes Alcanzados

El puntaje final y el porcentaje que ha obtenido cada tecnología de IaaS se obtiene de la siguiente manera:

$$\text{Puntaje Total del Análisis: } (Pt) = \sum(Pc)$$

$$\text{Puntaje Total de Eucalyptus: } (PtEus) = \sum(Peus)$$

$$\text{Puntaje Total de Nimbus: } (PtNis) = \sum(Pnis)$$

$$\text{Puntaje Total de OpenNebula: } (PtOpa) = \sum(Popa)$$

$$\text{Porcentaje Total de Eucalyptus: } (\%Eus) = \sum_1^5 \left( \frac{Peus(i)}{Pc(i)} * \% Pr(i) \right)$$

$$\text{Porcentaje Total de Nimbus: } (\%Nis) = \sum_1^5 \left( \frac{Pnis(i)}{Pc(i)} * \% Pr(i) \right)$$

$$\text{Porcentaje Total de OpenNebula: } (\%Opa) = \sum_1^5 \left( \frac{Popa(i)}{Pc(i)} * \% Pr(i) \right)$$

**Tabla III. XXXXXXIV Tabla General de Resultados**

Parámetro	Peso	Indicador	Eucalyptus	Nimbus	OpenNebula
<b>Configuración</b>	4	Grado de Complejidad	2	3	4
	4	Nivel de Conocimiento	3	4	4
	4	Cantidad de Tiempo	3	3	4
	4	Cantidad de Recurso Humano	3	3	3
<b>Total</b>	<b>16</b>		<b>11</b>	<b>13</b>	<b>15</b>
<b>Flexibilidad</b>	4	Cantidad Soporte Hypervisor	3	2	2
	4	Grado de Uso de DHCP	4	3	2
	4	Nivel de Despliegue	4	4	4
	4	Cantidad de Entornos de Programación	3	2	2
<b>Total</b>	<b>16</b>	<b>-</b>	<b>14</b>	<b>11</b>	<b>10</b>
<b>Gestión de Usuarios</b>	4	Creación, Modificación, Eliminación de Usuarios	3	2	4
	4	Monitoreo de Usuarios	4	4	4
	4	Protocolos de Seguridad	2	2	2
<b>Total</b>	<b>12</b>		<b>9</b>	<b>8</b>	<b>10</b>
<b>Gestión de Nodos</b>	4	Creación, Configuración y Eliminación de Nodos	4	2	4
	4	Monitoreo de Nodos	4	4	4
<b>Total</b>	<b>8</b>		<b>8</b>	<b>6</b>	<b>8</b>
<b>Gestión de Máquinas Virtuales</b>	4	Creación, Configuración y Eliminación de Máquinas Virtuales	4	4	2
	4	Iniciar, Reiniciar, Mostrar y Apagar Máquinas Virtuales	4	4	4
	4	Monitoreo de Máquinas Virtuales	4	4	4
	4	Variedad de Sistemas Operativos	2	2	2
<b>Total</b>	<b>16</b>	<b>-</b>	<b>14</b>	<b>14</b>	<b>12</b>
<b>Total</b>	<b>68</b>	<b>-</b>	<b>56</b>	<b>52</b>	<b>55</b>

**Fuente:** Roberto Carlos Salazar Gualoto

$$(Pt) = 16 + 16 + 12 + 8 + 16 = 68$$

$$(PtEus) = 11 + 14 + 9 + 8 + 14 = 56$$

$$(PtNis) = 13 + 11 + 8 + 6 + 14 = 52$$

$$(PtOpa) = 15 + 10 + 10 + 8 + 12 = 55$$

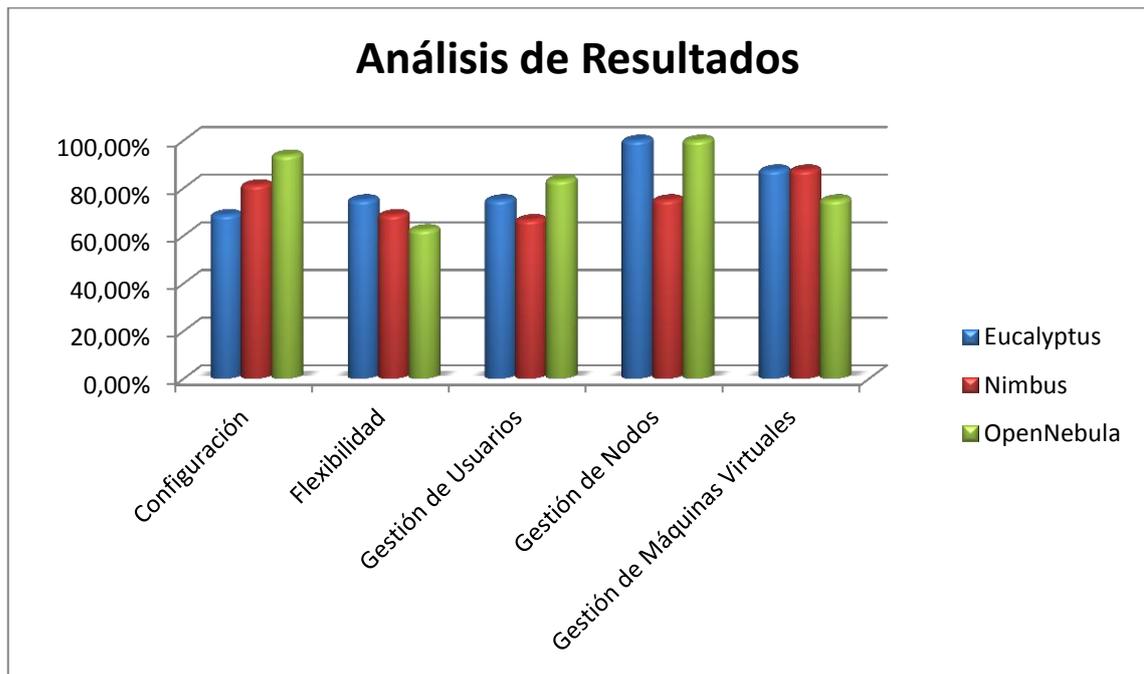


Figura III - 21. Comparación General de Porcentajes

$$\begin{aligned} (\%Eus) &= \left(\frac{11}{16}\right) * 20 + \left(\frac{14}{16}\right) * 20 + \left(\frac{9}{12}\right) * 10 + \left(\frac{8}{8}\right) * 20 + \left(\frac{14}{16}\right) * 30 \\ &= 85 \% \end{aligned}$$

**Equivalente a Muy Bueno**

$$\begin{aligned} (\%Nis) &= \left(\frac{13}{16}\right) * 20 + \left(\frac{11}{16}\right) * 20 + \left(\frac{8}{12}\right) * 10 + \left(\frac{6}{8}\right) * 20 + \left(\frac{14}{16}\right) * 30 \\ &= 77,91 \% \end{aligned}$$

**Equivalente a Bueno**

$$\begin{aligned} (\%Opa) &= \left(\frac{15}{16}\right) * 20 + \left(\frac{10}{16}\right) * 20 + \left(\frac{10}{12}\right) * 10 + \left(\frac{8}{8}\right) * 20 + \left(\frac{12}{16}\right) * 30 \\ &= 82,08 \% \end{aligned}$$

**Equivalente a Muy Bueno**

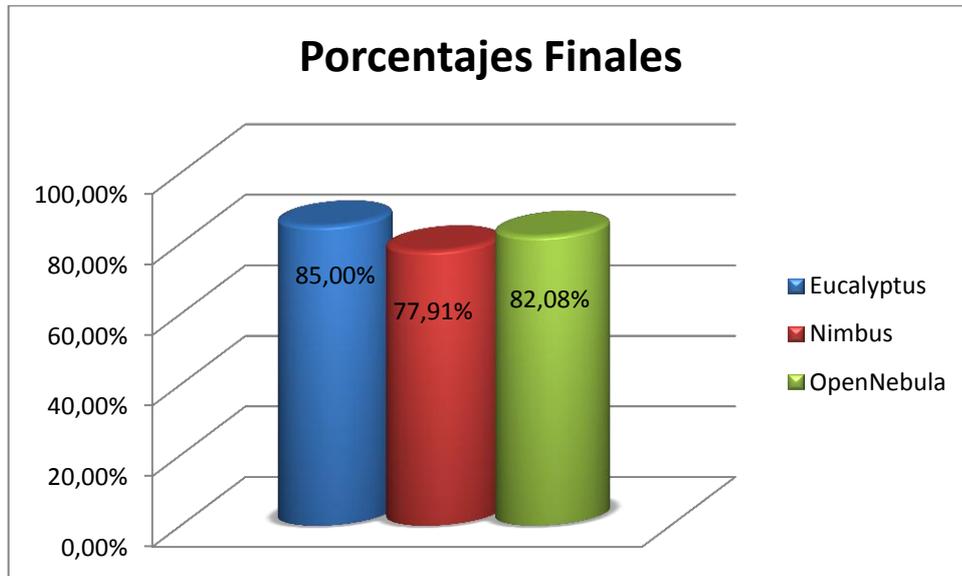


Figura III - 22. Comparación Final de Porcentajes

### 3.7.7. Interpretación

Como resultado del análisis y de acuerdo al porcentaje obtenido por cada una de las variables se ha obtenido que la tecnología de IaaS con el mayor porcentaje 85% equivalente a Muy Bueno es Eucalyptus, seguido de OpenNebula con un porcentaje de 82,08% equivalente a Muy Bueno y finalmente Nimbus con un porcentaje del 77,91% equivalente a Bueno.

### 3.7.8. Resultado del Análisis

- ✓ La Configuración de las IaaS Nimbus y OpenNebula tienen un porcentaje relativamente alto. Eucalyptus presenta un porcentaje menor del 68,75%, brindando varias facilidades entre ellas mínima ejecución de comandos,

conocimientos básicos sobre varios aspectos de Linux, cantidad de tiempo y recurso humano relativamente pequeño.

- ✓ En lo que se refiere a flexibilidad Eucalyptus es superior debido que brinda soporte a tres Hypervisores XEN, KVM y VMWARE, como en el Uso del servicio de DHCP de manera automática sumado a ello la forma de despliegue de manera dinámica más los entornos de programación en los que han sido desarrollado Hibernate, Axis y Java, en comparación con las dos IaaS restantes que solo brinda soporte a dos hypervisores y las demás funcionalidades son limitadas.
  
- ✓ Nimbus y OpenNebula cuentan con una amplia gama de comandos para gestionar los Usuarios, en comparación con Eucalyptus que presenta una interfaz tanto WEB como CLI para la Gestión de Usuarios la mismas que proveen recursos necesarios para la creación, modificación, eliminación y monitoreo de usuarios así como credenciales de seguridad propias, reducidos en la mínima ejecución de comandos.
  
- ✓ Eucalyptus y OpenNebula presentan el mismo nivel para Gestión de Nodos poseen variedad de comandos para permitir la creación, modificación eliminación y monitoreo de nodos de forma fácil y entendible para los administradores de la IaaS. En comparación con Nimbus que presenta mayor complejidad en el momento de realizar las operaciones necesarias para el funcionamiento y administración del nodo.

- ✓ El pilar fundamental de la IaaS es el servicio a ofrecer, Eucalyptus y Nimbus presentan el mismo porcentaje para Gestionar Máquinas Virtuales, es decir los comandos propios de cada IaaS para crear, configurar, eliminar, iniciar, reiniciar, mostrar, apagar y monitorear máquinas virtuales solamente difieren en la forma única de realizar dichas operaciones. Las tres IaaS analizadas presentan la misma variedad en cuanto a Sistemas Operativos Linux a excepción de Eucalyptus que permite la ejecución de SO Windows previo a la creación de una imagen.

### **3.7.9. Conclusión**

En base a lo expuesto anteriormente y de acuerdo a los puntajes alcanzados por cada uno de los parámetros de evaluación se llegó a determinar que la tecnología de Infraestructura como Servicio Eucalyptus brinda las mejores prestaciones para entregar Infraestructura como servicio con un porcentaje del 85% equivalente a Muy Bueno.

### **3.7.10. Comprobación de Hipótesis**

La hipótesis planteada es:

**H1:** La Implementación de un modelo de Infraestructura como Servicio Cloud Computing para Nubes Privadas permitirá mejorar la eficiencia de servicios computacionales de la EIS.

**Tabla III. XXXXXXV Parámetros de Hipótesis**

Variable	Parámetro	Indicador	Técnica	Herramienta
<b>Eficiencia</b>	Configuración	Grado de Complejidad	Monitoreo Observación de Ambientes de Prueba	Ficha de Observación
		Nivel de Conocimiento	Monitoreo Observación de Ambientes de Prueba	Ficha de Observación
		Cantidad de Tiempo	Monitoreo Observación de Ambientes de Prueba	Ficha de Observación
		Cantidad de Recurso Humano	Monitoreo Observación de Ambientes de Prueba	Ficha de Observación
	Flexibilidad	Cantidad Soporte Hypervisor	Investigación Bibliográfica	Ficha de Observación
		Grado de Uso de DHCP	Monitoreo Observación de Ambientes de Prueba	Ficha de Observación
		Nivel de Despliegue	Investigación Bibliográfica	Ficha de Observación
		Cantidad de Entornos de Programación	Investigación Bibliográfica	Ficha de Observación
	Gestión de Usuarios	Creación, Modificación, Eliminación de Usuarios	Monitoreo Observación de Ambientes de Prueba	Interfaz CLI y WEB de la IaaS
		Monitoreo de Usuarios	Monitoreo Observación de Ambientes de Prueba	Interfaz CLI y WEB de la IaaS
		Protocolos de Seguridad	Monitoreo Observación de Ambientes de Prueba Investigación Bibliográfica	Interfaz CLI y WEB de la IaaS
	Gestión de Nodos	Creación, Configuración y Eliminación de Nodos	Monitoreo Observación de Ambientes de Prueba	Interfaz CLI y WEB de la IaaS

		Monitoreo de Nodos	Monitoreo Observación de Ambientes de Prueba	Interfaz CLI y WEB de la IaaS
	Gestión de Máquinas Virtuales	Creación, Configuración y Eliminación de Máquinas Virtuales	Monitoreo Observación de Ambientes de Prueba	Interfaz CLI y WEB de la IaaS
		Iniciar, Reiniciar, Mostrar y Apagar Máquinas Virtuales	Monitoreo Observación de Ambientes de Prueba	Interfaz CLI y WEB de la IaaS
		Monitoreo de Máquinas Virtuales	Monitoreo Observación de Ambientes de Prueba	Interfaz CLI y WEB de la IaaS
		Variedad de Sistemas Operativos	Monitoreo Observación de Ambientes de Prueba	Interfaz CLI y WEB de la IaaS

**Fuente:** Roberto Carlos Salazar Gualoto

**Tabla III. XXXXXXVI Valores Parámetros de Hipótesis**

Variable	Parámetro	Indicador	Peso	Sin IaaS			
				IaaS	Eucalyptus	Nimbus	OpenNebula
<b>Eficiencia</b>	Configuración	Grado de Complejidad	4	1	2	3	4
		Nivel de Conocimiento	4	1	3	4	4
		Cantidad de Tiempo	4	3	3	3	4
		Cantidad de Recurso Humano	4	4	3	3	3
	<b>Total</b>	-	<b>16</b>	<b>9</b>	<b>11</b>	<b>13</b>	<b>15</b>
	Flexibilidad	Cantidad Soporte Hypervisor	4	2	3	2	2
		Grado de Uso	4	4	4	3	2

		de DHCP					
		Nivel de Despliegue	4	1	4	4	4
		Cantidad de Entornos de Programación	4	1	3	2	2
<b>Total</b>		-	<b>16</b>	<b>8</b>	<b>14</b>	<b>11</b>	<b>10</b>
Gestión de Usuarios		Creación, Modificación, Eliminación de Usuarios	4	3	3	2	4
		Monitoreo de Usuarios	4	3	4	4	4
		Protocolos de Seguridad	4	1	2	2	2
<b>Total</b>		-	<b>12</b>	<b>7</b>	<b>9</b>	<b>8</b>	<b>10</b>
Gestión de Nodos		Creación, Configuración y Eliminación de Nodos	4	1	4	2	4
		Monitoreo de Nodos	4	2	4	4	4
<b>Total</b>		-	<b>8</b>	<b>3</b>	<b>8</b>	<b>6</b>	<b>8</b>
Gestión de Máquinas Virtuales		Creación, Configuración y Eliminación de Máquinas Virtuales	4	3	4	4	2
		Iniciar, Reiniciar, Mostrar y Apagar Máquinas Virtuales	4	3	4	4	4
		Monitoreo de Máquinas Virtuales	4	3	4	4	4
		Variedad de	4	2	2	2	2

		Sistemas Operativos					
	Total	-	16	11	14	14	12
	TOTAL	-	68	38	56	52	55

Fuente: Roberto Carlos Salazar Gualoto

$$\begin{aligned}
 (\%SlaaS) &= \left(\frac{9}{16}\right) * 20 + \left(\frac{8}{16}\right) * 20 + \left(\frac{7}{12}\right) * 10 + \left(\frac{3}{8}\right) * 20 + \left(\frac{11}{16}\right) * 30 \\
 &= 55,21 \% \qquad \qquad \qquad \textbf{Equivalente a Regular}
 \end{aligned}$$

$$\begin{aligned}
 (\%Eus) &= \left(\frac{11}{16}\right) * 20 + \left(\frac{14}{16}\right) * 20 + \left(\frac{9}{12}\right) * 10 + \left(\frac{8}{8}\right) * 20 + \left(\frac{14}{16}\right) * 30 \\
 &= 85 \% \qquad \qquad \qquad \textbf{Equivalente a Muy Bueno}
 \end{aligned}$$

$$\begin{aligned}
 (\%Nis) &= \left(\frac{13}{16}\right) * 20 + \left(\frac{11}{16}\right) * 20 + \left(\frac{8}{12}\right) * 10 + \left(\frac{6}{8}\right) * 20 + \left(\frac{14}{16}\right) * 30 \\
 &= 77,91 \% \qquad \qquad \qquad \textbf{Equivalente a Bueno}
 \end{aligned}$$

$$\begin{aligned}
 (\%Opa) &= \left(\frac{15}{16}\right) * 20 + \left(\frac{10}{16}\right) * 20 + \left(\frac{10}{12}\right) * 10 + \left(\frac{8}{8}\right) * 20 + \left(\frac{12}{16}\right) * 30 \\
 &= 82,08 \% \qquad \qquad \qquad \textbf{Equivalente a Muy Bueno}
 \end{aligned}$$

**Tabla III. XXXXXXVII Valores y Porcentajes Finales**

IaaS	Peso	Porcentaje
<b>Sin IaaS</b>	38	55,21%
<b>Euacalyptus</b>	56	85,00%
<b>Nimbus</b>	52	77,91%
<b>OpenNebula</b>	55	82,08%

Fuente: Roberto Carlos Salazar Gualoto



**Figura III - 23. Comparación Final Porcentajes Sin y Con IaaS**

Haciendo referencia a la Tabla III. XXXXXXVII y por observación directa se concluye que la hipótesis planteada: La Implementación de un modelo de Infraestructura como Servicio Cloud Computing para Nubes Privadas mejora la eficiencia de servicios computacionales de la EIS por la relación porcentual en base a la comparación de no existir una IaaS con la presencia de IaaS Eucalyptus 29,79%, Nimbus 22,70% y OpenNebula 26.87%, por lo tanto cualquier IaaS mejora la eficiencia determinando esto que H1 es verdadera.

## CAPÍTULO IV

### Implementación de Prototipo de Infraestructura como Servicio “EIS”

#### 4.1. Introducción

La implementación de un Prototipo de Infraestructura como Servicio para la EIS desarrollado sobre la tecnología IaaS Eucalyptus requiere el detalle de la instalación y configuración de las herramientas necesarias para el adecuado funcionamiento y despliegue de servicios computacionales a brindarse a los usuarios de la IaaS.

#### 4.2. Instalación y Administración Eucalyptus

##### 4.2.1. Requerimientos Hardware y Software

Tabla IV. XXXXXXVIII Requerimientos Hardware

HW	FROND END	NODO
RAM	2 GB	3GB o superior
CPU	Intel Core 2 Duo o superior	Intel I5 o superior
DISCO	250 GB o superior	250 GB o superior
RED	2Tarjetas de red 10/100/1000 o superior	2 Tarjetas de red 10/100/1000/10000 o superior

Fuente: Roberto Carlos Salazar Gualoto

**Tabla IV. XXXXXXIX Requerimientos Software**

SW	FROND END Y NODO
SO	Centos 5.8 o superior
IaaS	Eucalyptus para Frond End
SSH	Secure Shell Client
Eucalyptus	Version 3.1 o superior

**Fuente:** Roberto Carlos Salazar Gualoto

### **4.3. Instalación de IaaS**

#### **4.3.1. Instalación en el Frond End**

Para la instalación del Frond End se requiere como prerequisite la instalación del SO Centos 5.8 o superior (6)

##### **4.3.1.1. Instalación de Dependencias**

###### **4.3.1.1.1. Configuración del Repositorio YUM para Eucalyptus Frond End**

Crear el archivo euca.repo con la ayuda del Editor VI

```
# vi /etc/yum.repos.d/euca.repo
```

Añadimos las siguientes líneas para crear el repositorio de Eucalyptus

```
[euca]
name=Eucalyptus
baseurl= http://downloads.eucalyptus.com/software/eucalyptus/3.1/centos/5/x86_64/
enabled=1
```

### 4.3.1.2. Instalación de Eucalyptus

Una vez creado el repositorio vamos a ejecutar el siguiente comando para la instalación de la IaaS.

```
# yum install eucalyptus-cloud eucalyptus-cc eucalyptus-walrus eucalyptus-sc --nogpgcheck
```

Una vez instalados desde el repositorio procedemos a iniciar los servicios mediante la ejecución de los siguientes comandos

```
# /etc/init.d/eucalyptus-cloud start  
# /etc/init.d/eucalyptus-cc start
```

Al ejecutar los comandos mencionados los servicios se proceden a iniciarse y para finalmente ver la siguiente figura.

```
[root@seuca ~]# /etc/init.d/eucalyptus-cloud start  
Starting Eucalyptus services: walrus sc cloud done.  
[root@seuca ~]# /etc/init.d/eucalyptus-cc start  
Starting Eucalyptus cluster controller: done.
```

Figura IV – 24. Proceso de Iniciación de Servicio de Eucalyptus

### 4.3.1.3. Configuración de Eucalyptus

Aquí se indica como establecer la configuración inicial de Eucalyptus

#### 4.3.1.3.1. Registro de Walrus

Para el registro de Walrus vamos a ejecutar el siguiente comando al cual añadiremos el nombre de la partición, dirección IP y nombre del componente en este caso se recomienda poner el nombre de la partición unido con el nombre de host.

```
# /usr/sbin/euca-conf --register-walrus --partition walrus --host 172.30.104.250 --component walrus_seuca
```

#### **4.3.1.3.2. Registro del Cluster**

Para el registro del Cluster necesitamos añadir el nombre de la partición, dirección IP y nombre del componente en este caso se recomienda poner el nombre de la partición unido con el nombre de host y ejecutar el siguiente comando.

```
# /usr/sbin/euca_conf --register-cluster --partition cluster01 --host 172.30.104.250 --component cc_seuca
```

#### **4.3.1.3.3. Registro Storage Controller**

Para el registro de Storage Controller necesitamos especificar el nombre de la partición, dirección IP y nombre del componente en este caso se recomienda poner el nombre de la partición unido con el nombre de host y ejecutar el siguiente comando.

```
# /usr/sbin/euca_conf --register-sc --partition cluster01 --host 172.30.104.250 --component sc_seuca
```

En el caso de existir un firewall, el mismo debe permitir que los componentes de Eucalyptus se comuniquen entre sí. En el Front End deben estar abiertos y accesibles los siguientes puertos 8443, 8773, 8774 y 9001.

#### **4.3.1.4. Instalación de Herramientas de Usuario**

##### **4.3.1.4.1. Configuración de Repositorio**

Añadimos datos al repositorio creado anteriormente para ello ejecutamos el siguiente comando

```
# vi /etc/yum.repos.d/euca.repo
```

Y añadimos los siguientes datos mediante la ayuda del Editor VI

```
[euca2ools]
name=Euca2ools
baseurl= http://downloads.eucalyptus.com/software/euca2ools/2.1/centos/5/x86_64/
enabled=1
```

#### 4.3.1.4.2. Instalación Euca2ools herramientas para el Usuario

Para la instalación de euca2ools vamos a realizarlo desde el repositorio creado con el siguiente comando.

```
# yum install euca2ools --nogpgcheck
```

#### 4.3.1.4.3. Configuración

Mediante el uso del interfaz WEB configuramos las herramientas de usuario para lo cual vamos a descargar las credenciales en el botón Download New Credentials

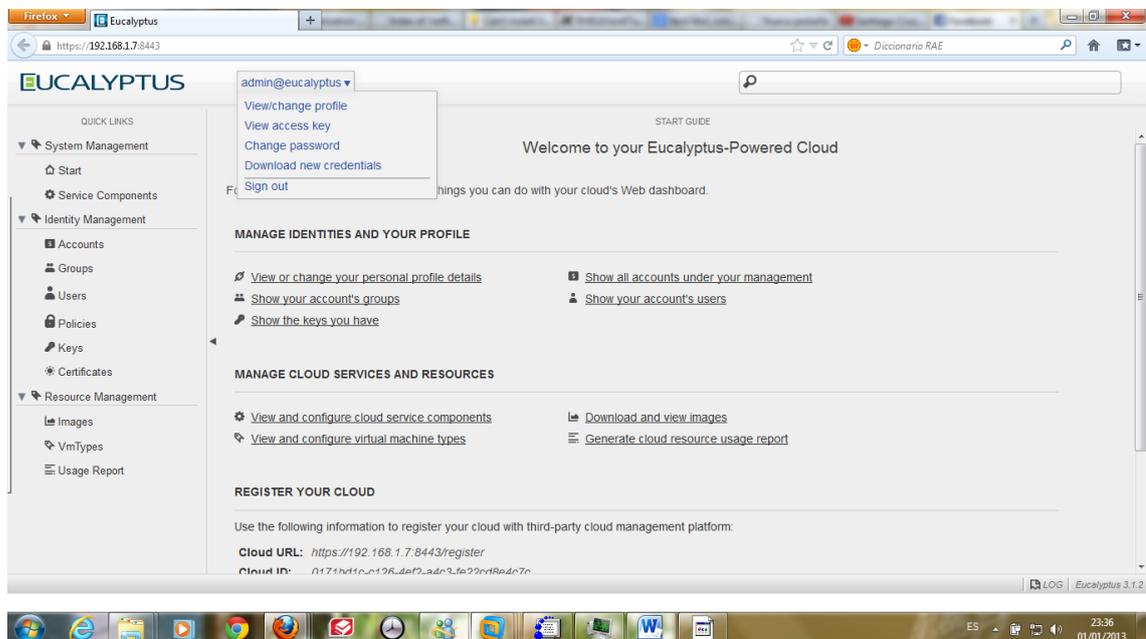


Figura IV – 25. Descarga de Credenciales

Al dar clic en el botón de descarga credenciales se descargara el par de claves pública y privada en un archivo llamado euca2-admin-x509.zip. Posteriormente tenemos que crear un directorio oculto llamado .euca ejecutando el siguiente comando

```
# mkdir ~/.euca
```

Descomprimos el archivo descargado ejecutando el siguiente comando

```
# unzip euca2-admin-x509.zip -d $HOME/.euca
```

Establecemos permisos para el directorio ejecutando el siguiente comando

```
# chmod 0700 ~/.euca
```

Establecemos permisos para los certificados

```
# chmod 0600 ~/.euca/*
```

Construimos las variables de entorno para que las herramientas del usuario puedan funcionar correctamente ejecutando el siguiente comando

```
# ~/.euca/eucarc
```

Actualizamos las variables globales usando el siguiente comando

```
# source ~/.euca/eucarc
```

Otra manera de realizar la descarga de credenciales es mediante el uso del siguiente comando

```
# /usr/sbin/euca_conf --get-credentials admin.zip
```

Descomprimos el archivo descargado ejecutando el siguiente comando

```
# unzip admin.zip -d $HOME/.euca
```

Establecemos permisos para el directorio ejecutando el siguiente comando

```
# chmod 0700 ~/.euca
```

Establecemos permisos para los certificados

```
# chmod 0600 ~/.euca/*
```

Construimos las variables de entorno para que las herramientas del usuario puedan funcionar correctamente ejecutando el siguiente comando

```
# ~/.euca/eucarc
```

Actualizamos las variables globales usando el siguiente comando

```
# source ~/.euca/eucarc
```

### **4.3.2. Instalación del Nodo XEN**

Previo a la instalación del nodo se debe tener instalado el SO Centos 5.8 o superior

#### **4.3.2.1. Instalación de Dependencias**

##### **4.3.2.1.1. Instalación de Hypervisor XEN**

Para instalar Xen desde el repositorio debemos ejecutar el siguiente comando (9)

```
# yum install xen kernel-xen
```

## Configurar el GRUB

Ahora tenemos que configurar el GRUB para que Centos inicie con la versión del Kernel modificado con XEN. Con la ayuda del editor VI modificamos el archivo *menu.lst* que tiene la configuración de boot del sistema mediante la ejecución del siguiente comando

```
# vi /boot/grub/menú.lst
```

Posteriormente cambiaremos el orden de boot

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-308.8.2.el5xen)
    root (hd0,0)
    kernel /xen.gz-2.6.18-308.8.2.el5
    module /vmlinuz-2.6.18-308.8.2.el5xen ro root=LABEL=/ rhgb quiet
    module /initrd-2.6.18-308.8.2.el5xen.img
title CentOS (2.6.18-308.8.2.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-308.8.2.el5 ro root=LABEL=/ rhgb quiet
initrd /initrd-2.6.18-308.8.2.el5.img
```

En este caso editamos el default = 0 para que inicie con la versión del kernel propia de XEN y procedemos a reiniciar el servidor ejecutando el siguiente comando para que los cambios en los archivos surtan efecto.

```
# restart
```

#### 4.3.2.1.2. Configurar XEN

Una vez reiniciado el servidor procedemos a configurar el archivo de XEN habilitando opciones con la ayuda del editor VI ejecutando el siguiente comando

```
# vi /etc/xen/xend-config.sxp
```

Aquí cambiamos las siguientes opciones

```
(xend-http-server yes)
(xend-unix-server yes)
```

Reiniciamos el servicio XEN

```
# /etc/init.d/xend restart
```

#### 4.3.2.1.3. Configuración del Repositorio YUM para Eucalyptus Nodo

Crear el archivo euca.repo con la ayuda del Editor VI

```
# vi /etc/yum.repos.d/euca.repo
```

Añadimos las siguientes líneas para crear el repositorio de Eucalyptus

```
[euca]
name=Eucalyptus
baseurl=http://downloads.eucalyptus.com/software/eucalyptus/3.1/centos/5/x86_64/
enabled=1
```

#### 4.3.3. Instalación de Eucalyptus en el Nodo

Para la instalación del controlador del nodo desde repositorio debemos ejecutar el siguiente comando

```
# yum install eucalyptus-nc --nogpgcheck
```

Posteriormente procedemos a iniciar el node controller ejecutando el siguiente comando

```
# /etc/init.d/eucalyptus-nc start
```

Una vez ejecutado el comando se presentará la siguiente figura en la que indica que el servicio se ha iniciado correctamente.

```
[root@snceuca ~]# /etc/init.d/eucalyptus-nc start
You should have at least 32 loop devices
Starting Eucalyptus services:
Enabling bridge netfiltering for eucalyptus.
done.
[root@snceuca ~]#
```

**Figura IV – 26. Inicio Servicio Node Controller**

Una vez realizada la configuración e inicio del servicio procedemos a verificar la creación del usuario eucalyptus y la posibilidad de interactuar con libvirt en los nodos de computación ejecutando el siguiente comando.

```
# su eucalyptus -c "virsh list"
```

Una vez ejecutado el comando si todo este bien se nos presentara la siguiente figura

```
[root@snceuca ~]# su eucalyptus -c "virsh list"
Id Nombre          Estado
-----
0 Domain-0         ejecutando
```

**Figura IV – 27. Comprobación de iteración de libvirt y el nodo de computación**

En el caso de existir firewall debe permitir el acceso al puerto 8475 para que exista comunicación con el nodo.

### 4.3.3.1. Registro de Node Controller

Para el registro de node controller necesitamos especificar la dirección IP del nodo este comando lo ejecutamos en el Front End.

```
# /usr/sbin/euca_conf --register-nodes 172.30.104.252
```

## 4.4. Administración de Eucalyptus

### 4.4.1. Registrar el Kernel

Para registrar el kernel procedemos a por medio de la página WEB de imágenes de Eucalyptus descargar una imagen certificada.

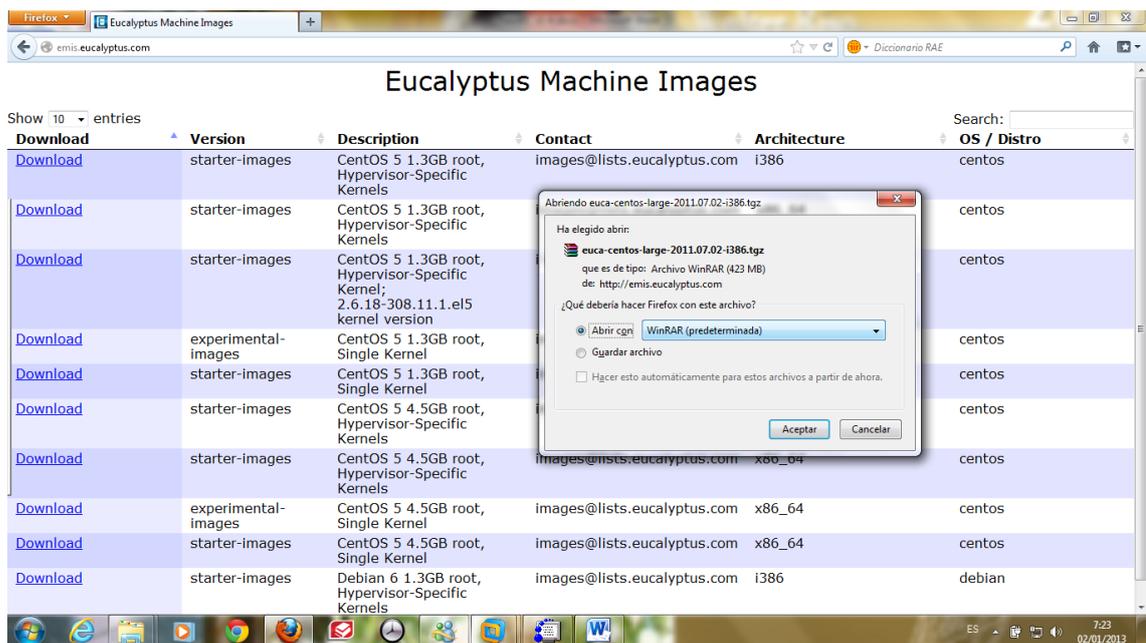


Figura IV – 28. Descarga Imagen Certificada de Centos

Una vez descargado el archivo de nombre *euca-centos-large-2011.07.02-i386.tgz* procedemos a descomprimirlo ejecutando el siguiente comando.

```
# tar -zxvf euca-centos-large-2011.07.02-i386.tgz
```

Ahora vamos a proceder a registrar el kernel empaquetando el archivo de kernel en este caso *vmlinuz-2.6.24-19-xen* y añadiendo a que arquitectura pertenece, ejecutando el siguiente comando

```
# euca-bundle-image -i euca-centos-large-2011.07.02-i386/xen-kernel/vmlinuz-2.6.24-19-xen --kernel true -r i386
```

Ahora vamos a subir la imagen de kernel a el Repositorio Walrus ejecutando el siguiente comando para ello necesitamos el nombre de archivo de bucler del kernel en este caso *cen5\_3\_i386-kernel-bucket* y el archivo creado al empaquetar el kernel *vmlinuz-2.6.24-19-xen.manifest.xml*.

```
# euca-upload-bundle -b cen5_4_i386_kernel_bucket -m /tmp/vmlinuz-2.6.24-19-xen.manifest.xml
```

Posteriormente vamos a registrar la imagen adicionando la arquitectura a la cual pertenece ejecutando el siguiente comando

```
# euca-register cen5_4_i386_image_bucket/centos.5-3.x86.img.manifest.xml -a i386
```

De esta manera queda registrada la imagen de kernel podemos verificar el registro en la interfaz WEB de la IaaS tal como se muestra en la siguiente figura

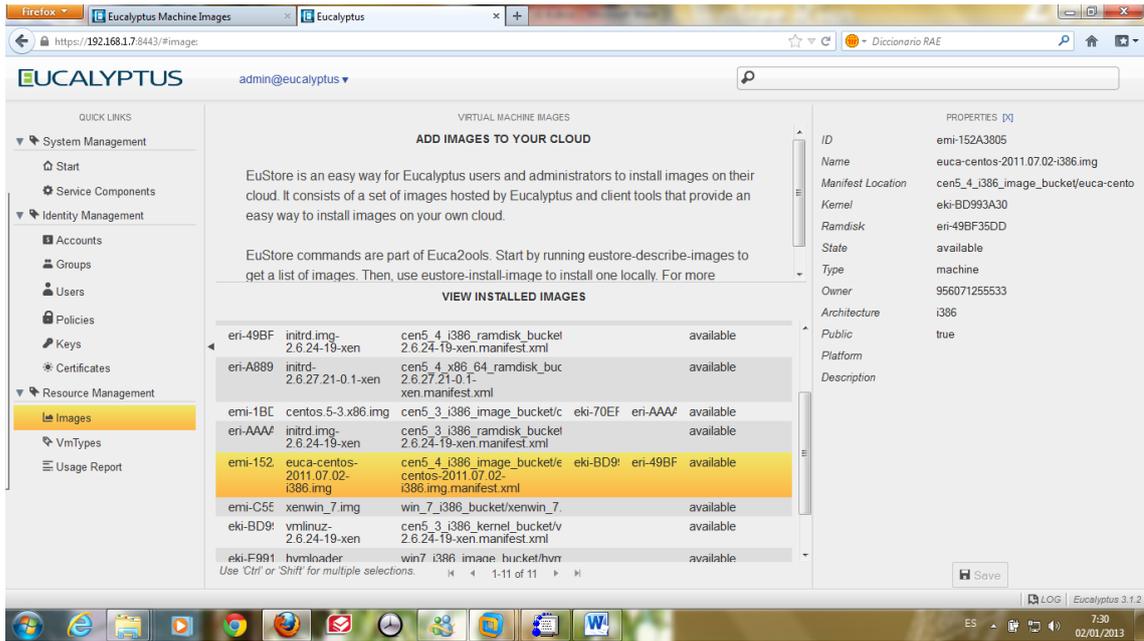


Figura IV – 29. Registro de kernel de la Imagen Certificada de Centos

#### 4.4.2. Registro Ramdisk

Para registrar la ramdisk debemos empaquetar el archivo `initrd` ejecutando el siguiente comando el cual nos genera un archivo llamado `manifest.xml` para ello se adicionara la arquitectura a la cual pertenece

```
# euca-bundle-image -i euca-centos-large-2011.07.02-i386/xen-kernel/initrd.  
img-2.6.24-19-xen --ramdisk true -r i386
```

Subimos la imagen de Ramdisk al repositorio Walrus ejecutando el siguiente comando para ello se necesita especificar el nombre de la reserva en este caso `cen5_4_i386-ramdisk.bucket` y el archivo generado al empaquetar el `initrd` `initrd.img-2.6.24-19-xen.manifest.xml`

```
# euca-upload-bundle -b cen5_4_i386_ramdisk_bucket -m /tmp/initrd.img-2.6.2  
4-19-xen.manifest.xml
```

Posteriormente registramos la ramdisk en Eucalyptus para ello necesitamos el nombre de la bucket de reserva en este caso *centos-ramdisk.bucket/initrd.img-2.6.24-19-xen.manifest.xml* y la arquitectura a la cual pertenece en este caso i386

```
# euca-register cen5_4_i386_ramdisk_bucket/initrd.img-2.6.24-19-xen.manifest.xml -a i386
```

Una vez realizada el registro de la ramdisk procedemos a verificar la existencia de la misma en la interfaz WEB tal como muestra la siguiente figura.

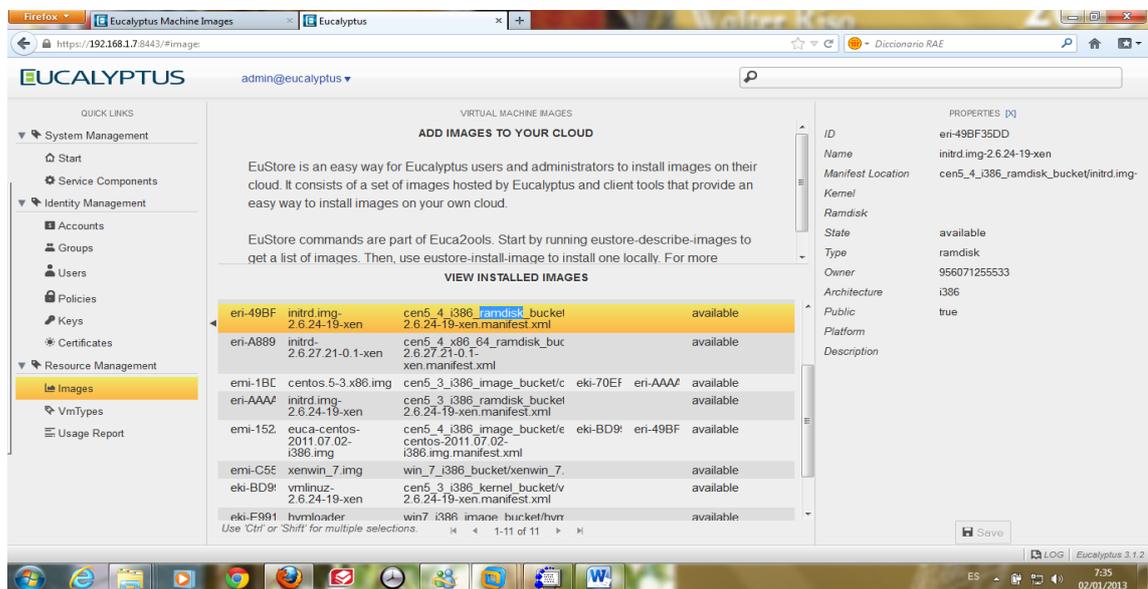


Figura IV – 30. Registro de ramdisk de la Imagen Certificada de Centos

### 4.4.3. Gestión de Imágenes

#### 4.4.3.1. Registro de Imagen Centos

Para registrar la imagen de centos debemos empaquetar el archivo imagen en este caso euca-centos-2011.07.02-i386.img ejecutando el siguiente comando y además asociamos los identificadores del kernel, ramdisk registrados anteriormente y la arquitectura a la cual pertenecen.

```
# euca-bundle-image -i euca-centos-large-2011.07.02-i386/euca-centos-2011.07.02-i386.img --ramdisk eri-49BF35DD --kernel eki-BD993A30 -r i386
```

Una vez empaquetada la imagen y generado el archivo *euca-centos-2011.07.02-i386.img.manifest.xml* procedemos a subir la imagen de disco a Walrus especificando el nombre de reserva en este caso *cen5\_4\_i386\_image\_bucket* y el archivo imagen que generamos al empaquetar, para lo cual ejecutamos el siguiente comando

```
# euca-upload-bundle -b cen5_4_i386_image_bucket -m /tmp/euca-centos-2011.07.02-i386.img.manifest.xml
```

Posteriormente registraremos la imagen certificada de Centos en Eucalyptus especificando el nombre de la reserva para el caso *cen5\_4\_i386\_image\_bucket/euca-centos-2011.07.02-i386.img.manifest.xml* y la arquitectura *i386* ejecutando el siguiente comando.

```
# euca-register cen5_4_i386_image_bucket/euca-centos-2011.07.02-i386.img.manifest.xml -a i386
```

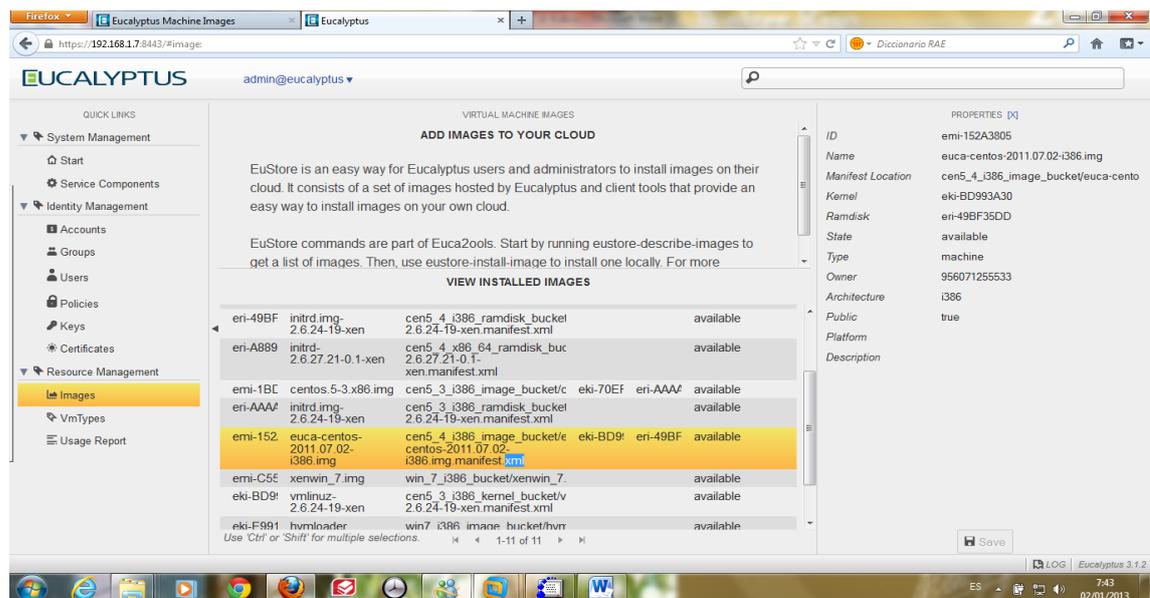


Figura IV – 31. Registro de la Imagen Certificada de Centos

Una vez registrada la imagen certificada de Centos vamos a verificar su registro en la interfaz WEB, sino existió ningún error la figura previa se nos presentará.

#### 4.4.3.2. Eliminación de Imagen

Para la eliminación de una imagen anteriormente registrada se ejecutara el siguiente comando

```
# euca-delete-bundle -b centos-image-bucket
```

#### 4.4.3.3. Descargar Imagen desde Repositorio

Para descargar una imagen desde el repositorio basta con ejecutar el siguiente comando y la imagen descargada se guardara en el directorio /tmp

```
# euca-download-bundle -b centos-image-bucket
```

#### 4.4.3.4. Mostrar Información de las Imágenes Registradas

Para mostrar información de las imágenes registradas ejecutaremos el siguiente comando

```
# euca-describe-images
```

Posteriormente se nos presentará la siguiente figura que describe las imágenes registradas información congruente a la presentada en la interfaz WEB

```
956071255533    available    public      i386    machine eki-70EF3A19 e
ri-AAAA36EB          instance-store
IMAGE emi-152A3805    cen5_4_i386_image_bucket/euca-centos-2011.07.02-i386.img
.manifest.xml 956071255533    available    public    i386    machinee
ki-BD993A30    eri-49BF35DD          instance-store
IMAGE emi-C55F37CE    win_7_i386_bucket/xenwin_7.img.manifest.xml    95607125
5533    available    public      i386    machine    i
nstance-store
IMAGE eki-BD993A30    cen5_3_i386_kernel_bucket/vmlinuz-2.6.24-19-xen.manifest
.xml    956071255533    available    public    i386    kernel    i
nstance-store
IMAGE eki-70EF3A19    cen5_3_i386_kernel_bucket/vmlinuz-2.6.24-19-xen.manifest
.xml    0000000000001    available    public    i386    kernel    i
nstance-store
```

Figura IV – 32. Listado de Imágenes Registradas

#### 4.4.4. Gestión de Máquinas Virtuales

##### 4.4.4.1. Creación de Keypairs

Antes de iniciar una máquina virtual tenemos la necesidad de crear claves para el inicio de una imagen certificada para ello ejecutaremos el siguiente comando

```
# euca-add-keypair mykey-euca > mykey-euca.private
```

Una vez creadas las claves procedemos a cambiar los permisos para tener mayor seguridad ejecutando el siguiente comando.

```
# chmod 0600 mykey-euca.private
```

##### 4.4.4.2. Creación Máquina Virtual con SO Centos

Para iniciar una máquina virtual tenemos que ejecutar el siguiente comando con los siguientes parámetros nombre del fichero de claves, identificador de imagen, tipo de instancia.

```
# euca-run-instances emi-152A3805 -k mykey_euca -t m1.large
```

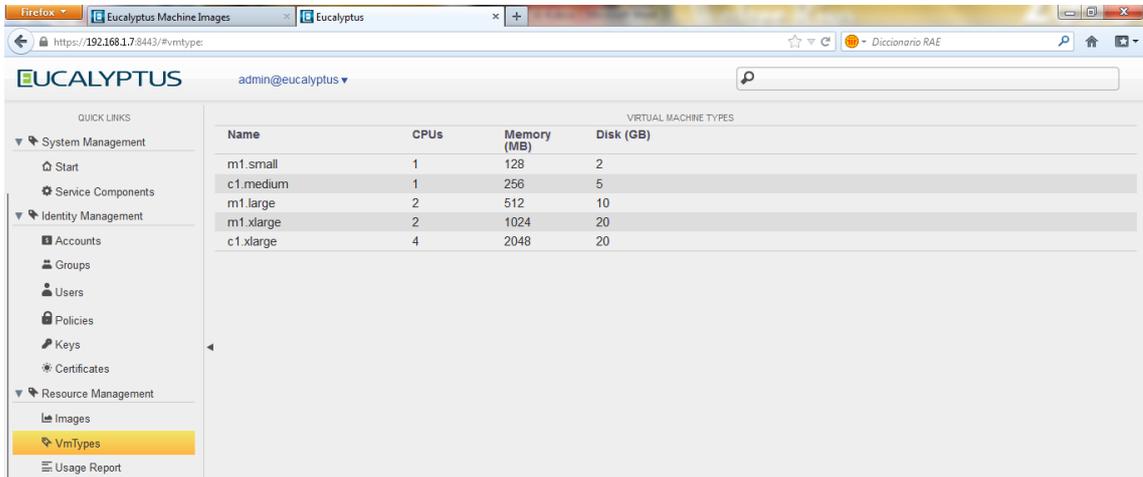
Si no existe ningún error se nos presentará la siguiente figura en la que se nos muestra información acerca de la máquina creada

```
[root@seuca ~]# euca-run-instances emi-152A3805 -k mykey_euca -t m1.large
RESERVATION   r-COCE4799      956071255533    default
INSTANCE      i-8C32489E      emi-152A3805    0.0.0.0 0.0.0.0 pending mykey_eu
ca           0                m1.large        2013-01-02T12:50:49.983Z    cluster0
1            eki-BD993A30    eri-49BF35DD    monitoring-disabled        0.0.0.00
.0.0.0       instance-store
[root@seuca ~]#
```

Figura IV – 33. Información de Máquinas Virtuales Creada

## Tipos de Máquinas Virtuales

Las tipos de máquinas virtuales que dispone la IaaS Eucalyptus se presenta en la siguiente imagen.



The screenshot shows the Eucalyptus web interface. On the left is a navigation menu with categories like System Management, Identity Management, and Resource Management. The 'VmTypes' option is highlighted. The main content area displays a table of virtual machine types.

Name	CPUs	Memory (MB)	Disk (GB)
m1.small	1	128	2
c1.medium	1	256	5
m1.large	2	512	10
m1.xlarge	2	1024	20
c1.xlarge	4	2048	20

Figura IV – 34. Tipos de Máquinas Virtuales

### 4.4.4.3. Iniciar Máquina Virtual Centos

Para iniciar la máquina virtual previamente creada vamos a ejecutar el siguiente comando añadiendo el nombre del fichero de claves creadas.

```
# ssh -i mykey.private 172.30.104.168
```

Posteriormente se nos presentará la siguiente figura con la ejecución de la máquina virtual en la cual se puede realizar cualquier tarea deseada

```
[root@seuca ~]# ssh -i mykey_euca.private 172.30.104.168
The authenticity of host '172.30.104.168 (172.30.104.168)' can't be established.
RSA key fingerprint is d9:53:41:68:42:91:9a:83:3e:5e:af:72:20:7a:f3:08.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.30.104.168' (RSA) to the list of known hosts.
-bash-3.2# uname -a
Linux localhost 2.6.24-19-xen #2 SMP Fri May 23 03:11:08 JST 2008 i686 i686 i386
GNU/Linux
-bash-3.2#
```

Figura IV – 35. Inicio de Máquina Virtual

#### 4.4.4.4. Reinicio de Máquinas Virtuales

Para reiniciar una máquina virtual ejecutaremos el siguiente comando anexando el identificador de la instancia

```
# euca-reboot-instances i-22364047
```

#### 4.4.4.5. Mostrar Máquinas Virtuales en Ejecución

Para mostrar las máquinas virtuales en ejecución se ejecutara el siguiente comando

```
# euca-describe-instances
```

Posteriormente se presentará la siguiente figura que muestra el listado de máquinas virtuales en ejecución

```
[root@seuca ~]# euca-describe-instances
RESERVATION    r-53884230      956071255533   default
INSTANCE       i-22364047      emi-152A3805   172.30.104.168 172.30.104.168 r
unning mykey_euca    0              m1.large       2013-01-02T21:38:26.522Z
cluster01      eki-BD993A30    eri-49BF35DD   monitoring-disabled 1
72.30.104.168  172.30.104.168                instance-store
[root@seuca ~]#
```

Figura IV – 36. Listado de Máquinas Virtuales en ejecución

#### 4.4.4.6. Monitorear Máquina Virtual

Para el monitoreo de máquinas virtuales ejecutaremos el siguiente comando

```
# euca-describe-instances i-22364047
```

Posteriormente se nos presentará la siguiente figura en la que se detalla información acerca del estado de la máquina virtual solicitada

```
[root@seuca ~]# euca-describe-instances i-22364047
RESERVATION    r-53884230      956071255533   default
INSTANCE       i-22364047      emi-152A3805   172.30.104.168 172.30.104.168 r
unning mykey_euca    0              m1.large       2013-01-02T21:38:26.522Z
cluster01      eki-BD993A30    eri-49BF35DD   monitoring-disabled 1
72.30.104.168  172.30.104.168                instance-store
[root@seuca ~]#
```

Figura IV – 37. Monitoreo Máquina Virtual

#### 4.4.4.7. Apagar Máquinas Virtuales

Para apagar la máquina virtual debemos ejecutar el siguiente comando anexando el identificador de la máquina virtual.

```
# euca-terminate-instances i-22364047
```

Posteriormente la máquina virtual se apaga y el estado pasara a ser de shut down y se nos presentará la siguiente imagen

```
[root@seuca ~]# euca-terminate-instances i-22364047
INSTANCE          i-22364047
[root@seuca ~]# euca-describe-instances i-22364047
RESERVATION      r-53884230      956071255533    default
INSTANCE         i-22364047      emi-152A3805    172.30.104.168  172.30.104.168 s
hutting-down    mykey_euca      0               ml.large        2013-01-02T21:38
:26.522Z        cluster01       eki-BD993A30    eri-49BF35DD    monitori
ng-disabled     172.30.104.168 172.30.104.168 instance-store
[root@seuca ~]#
```

Figura IV – 38. Apagar Máquina Virtual

#### 4.4.4.8. Creación de Imagen Windows

La primera tarea para instalar el sistema operativo base de Windows usando Xen es:

Detener el servicio de eucaliptus-nc en el nodo ejecutando el siguiente comando.

```
# /etc/init.d/eucaliptus-nc stop
```

Crear un archivo de disco en blanco que sirve de base para la imagen del VM usando el siguiente comando

```
# dd if=/dev/zero of=xenwin_7.img bs=1M count=1 seek=10000
```

Con la ayuda del editor VI procedemos a crear un archivo llamado maquina en donde especificamos las opciones para la creación e instalación de la VM.

```
# vi maquina
```

Añadimos lo siguiente que describe varias opciones de configuración como el kernel la cantidad de memoria imagen de disco imagen de sistema operativo y otras opciones.

```
kernel = "/usr/lib/xen/boot/hvmloader"  
builder = 'hvm'  
device_model = '/usr/lib/xen/bin/qemu-dm'  
memory = 1024  
name = "win_7"  
vcpus = 1  
vif = [ 'type=ioemu, bridge=xenbr0' ]  
disk = ['file:/root/Win_7.img,ioemu:hda,w','file:/root/win7.iso,ioemu:hdc:cdrom,r']  
boot = 'dc'  
ne2000 = 0  
acpi= 1  
apic= 1  
dhcp='dhcp'  
vnc=0  
vnclisten='0.0.0.0'  
vncunused=0  
vncviewer=0  
sdl=1  
nographic=0  
usb=1  
usbdevice='tablet'  
pae=1
```

Iniciamos ejecutando el siguiente comando

```
xm create maquina
```

Una vez iniciada la VM procedemos a instalar el sistema operativo base Windows 7, de la misma manera que se realiza en una máquina física, luego lo actualizamos y posteriormente se instala el complemento de integración con la IaaS.

#### 4.4.4.8.1. Instalación Complemento de Integración Eucalyptus Windows

Para instalar el complemento que permite el funcionamiento de la imagen de Windows con Eucalyptus se procede a descargar de la siguiente página <http://downloads.eucalyptus.com/software/tools/windows-prep/> la herramienta llamada *windows-prep-tools-3.1.0.iso*

Una vez que se ha descargado el paquete se procede a cambiar la siguiente línea

```
disk = ['file:/root/Win_7.img,ioemu:hda,w','file:/root/windows-prep-tools-3.1.0.iso,ioemu:hdc:cdrom,r']
```

Iniciamos la VM y procedemos a abrir la unidad de DVD y buscamos el archivo de nombre EucalyptusWindowsIntegration.msi damos clic

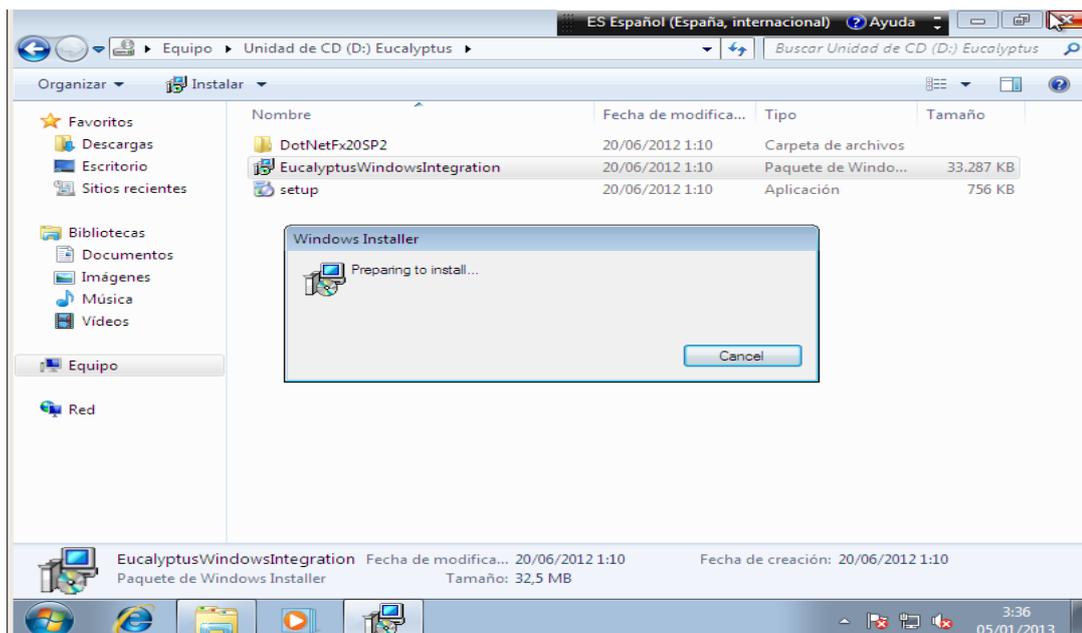
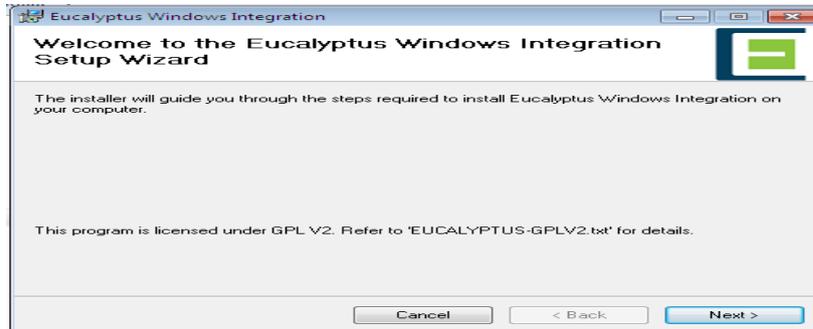


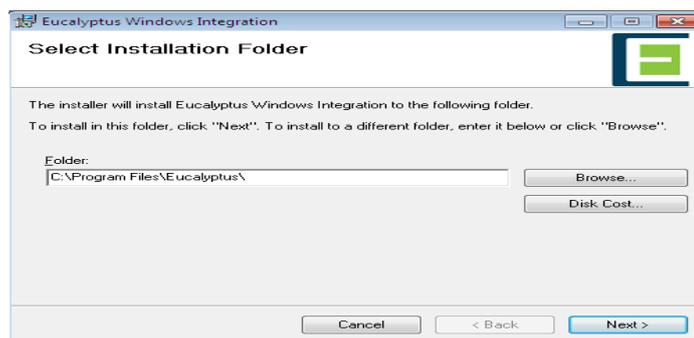
Figura IV – 39. Preparar Instalación Eucalyptus Integration Windows

Cuando el archivo termine de preparar la instalación nos mostrará la siguiente imagen en la que se debe dar clic en Next para iniciar la instalación.



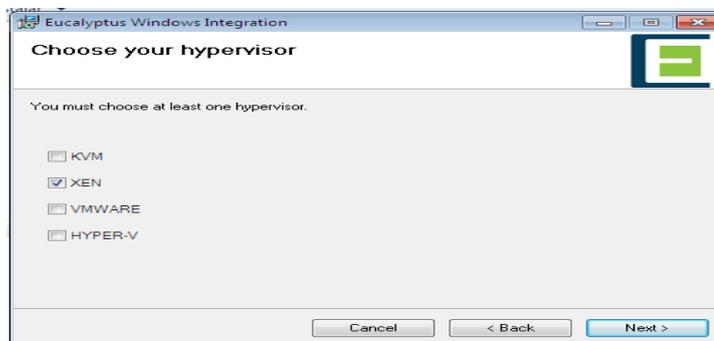
**Figura IV – 40. Inicio Instalación Eucalyptus Integration Windows**

Escogemos la ubicación donde se van a instalar los archivos de configuración del complemento se recomienda dejarlo en la ubicación por defecto



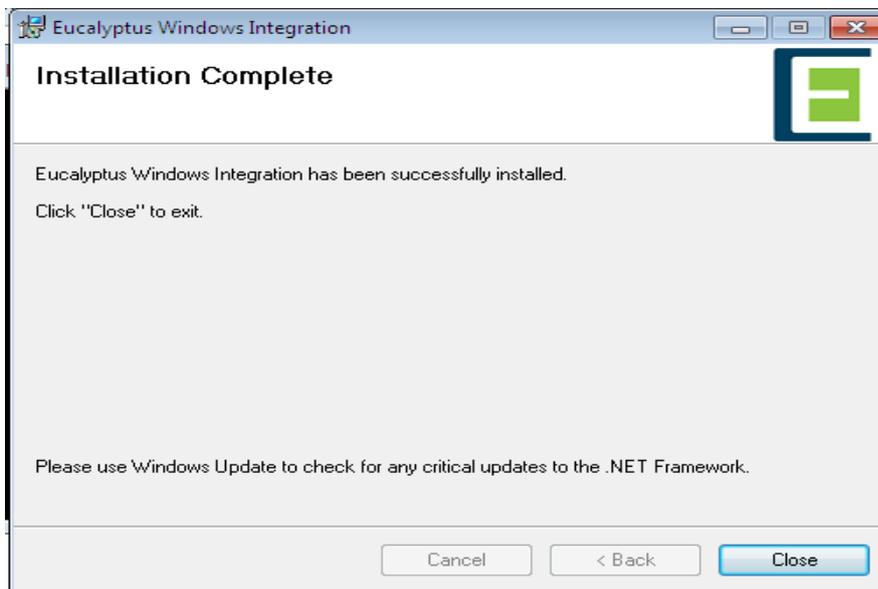
**Figura IV – 41. Ubicación Archivo Eucalyptus Integration Windows**

En la siguiente pantalla se escoge el hypervisor con el cual se está trabajando



**Figura IV – 42. Hypervisor Eucalyptus Integration Windows**

Confirmamos la instalación y esperamos a que finalice correctamente como muestra la siguiente pantalla



**Figura IV – 43. Instalación Correcta Eucalyptus Integration Windows**

#### **4.4.4.8.2. Registro de Imagen Windows**

Para registrar la imagen de Windows debemos empaquetar el archivo de la imagen ejemplo xenwin\_7.img ejecutando el siguiente comando, en este caso esta imagen no necesita kernel ni ramdisk asociados solo es necesario especificar la arquitectura a la que pertenece.

```
# euca-bundle-image -i xenwin_7.img -r i386
```

Este proceso tardara un poco de tiempo hasta generar el archivo xml tal como se muestra en la siguiente figura.

```
[root@seuca ~]# euca-bundle-image -i xenwin_7.img -r i386
Checking image
Compressing image
Encrypting image
Splitting image...
Part: xenwin_7.img.part.00
Part: xenwin_7.img.part.01
Part: xenwin_7.img.part.02
Part: xenwin_7.img.part.03
Part: xenwin_7.img.part.04
Part: xenwin_7.img.part.05
Part: xenwin_7.img.part.06
Part: xenwin_7.img.part.07
Part: xenwin_7.img.part.08
Part: xenwin_7.img.part.09
Part: xenwin_7.img.part.236
Part: xenwin_7.img.part.237
Part: xenwin_7.img.part.238
Part: xenwin_7.img.part.239
Part: xenwin_7.img.part.240
Part: xenwin_7.img.part.241
Part: xenwin_7.img.part.242
Part: xenwin_7.img.part.243
Generating manifest /tmp/xenwin_7.img.manifest.xml
[root@seuca ~]#
```

Figura IV – 44. Registro de Imagen Windows

Una vez empaquetada la imagen y generado el archivo xenwin\_7.img.manifest.xml procedemos a subir la imagen de disco a Walrus especificando el nombre de reserva en este caso win7\_i386\_image\_bucket y el archivo imagen que generamos al empaquetar, para lo cual ejecutamos el siguiente comando

```
# euca-upload-bundle -b win7_i386_image_bucket -m /tmp/
xenwin_7.img.manifest.xml
```

Este proceso también demora unos minutos para que se presente la siguiente figura

```
[root@seuca ~]# euca-upload-bundle -b win7_i386_image_bucket -m /tmp/xenwin_7.im
g.manifest.xml
Checking bucket: win7_i386_image_bucket
Creating bucket: win7_i386_image_bucket
Uploading manifest file
Uploading part: xenwin_7.img.part.00
Uploading part: xenwin_7.img.part.01
Uploading part: xenwin_7.img.part.02
Uploading part: xenwin_7.img.part.03
Uploading part: xenwin_7.img.part.04
Uploading part: xenwin_7.img.part.05
Uploading part: xenwin_7.img.part.06
Uploading part: xenwin_7.img.part.07
Uploading part: xenwin_7.img.part.08
Uploading part: xenwin_7.img.part.09
Uploading part: xenwin_7.img.part.10
Uploading part: xenwin_7.img.part.230
Uploading part: xenwin_7.img.part.231
Uploading part: xenwin_7.img.part.232
Uploading part: xenwin_7.img.part.233
Uploading part: xenwin_7.img.part.234
Uploading part: xenwin_7.img.part.235
Uploading part: xenwin_7.img.part.236
Uploading part: xenwin_7.img.part.237
Uploading part: xenwin_7.img.part.238
Uploading part: xenwin_7.img.part.239
Uploading part: xenwin_7.img.part.240
Uploading part: xenwin_7.img.part.241
Uploading part: xenwin_7.img.part.242
Uploading part: xenwin_7.img.part.243
Uploaded image as win7_i386_image_bucket/xenwin_7.img.manifest.xml
[root@seuca ~]#
```

Figura IV – 45. Registro bucket de Imagen Windows

Posteriormente registramos la imagen certificada de Windows en Eucalyptus especificando el nombre de la reserva para el caso win7\_i386\_image\_bucket/xenwin\_7.img.manifest.xml ejecutando el siguiente comando.

```
# euca-register win7_i386_image_bucket/ xenwin_7.img.manifest.xml
```

Una vez registrada la imagen se nos presentará en el entorno web de Eucalyptus la información de la imagen registrada tal como se visualiza en la figura siguiente.

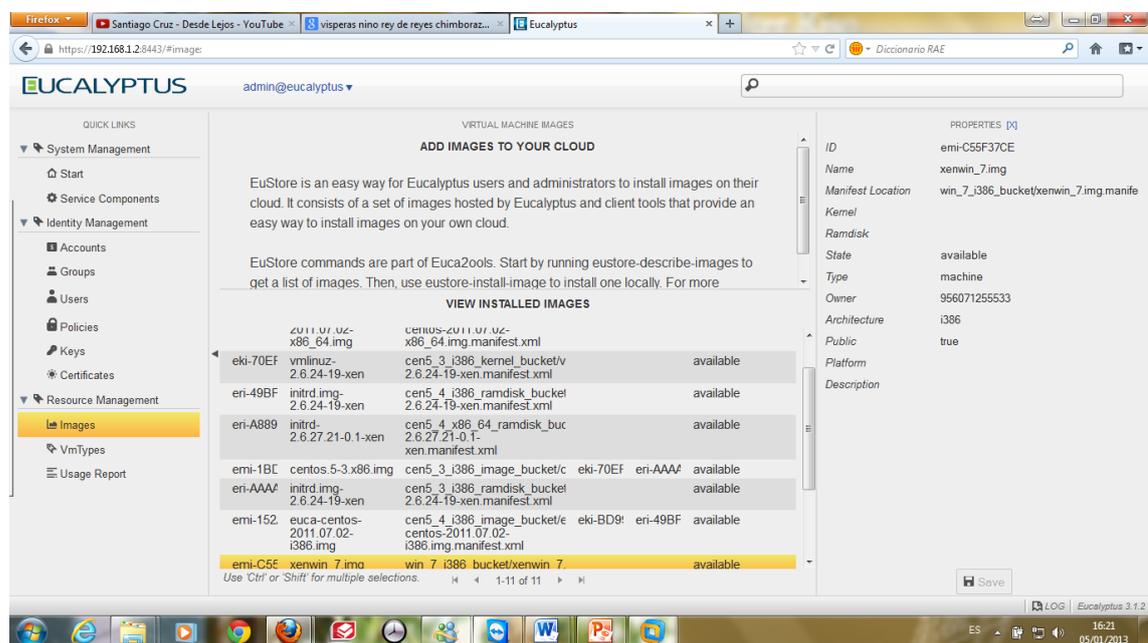


Figura IV – 46. Información Imagen Windows

#### 4.4.4.8.3. Iniciar Máquina Virtual Windows

Para iniciar la VM de Windows se debe ejecutar el siguiente comando y adicionalmente adjuntar el tipo de VM en este caso *m1.xlarge*

```
# euca-run-instances emi-C55F37CE -t m1.xlarge
```

Una vez que se ha ejecutado el comando se presentará la siguiente figura

```
[root@seuca ~]# euca-run-instances emi-C55F37CE -t m1.xlarge
RESERVATION    r-954443D5    956071255533    default
INSTANCE      i-3CF24665    emi-C55F37CE    0.0.0.0 0.0.0.0 pending    0
m1.xlarge     2013-01-05T21:31:52.662Z    cluster01     eki-70EF3A19    m
onitoring-disabled    0.0.0.0 0.0.0.0    instance-store
[root@seuca ~]#
```

Figura IV – 47. Iniciar VM Windows

Cuando la VM se instancie ejecute o despliegue podremos hacer conexión de Escritorio Remoto de la como se muestra en la figura a continuación.



Figura IV – 48. Conexión remoto VM Window

De este modo se realiza la conexión remota la cual nos pide se realice la autenticación usando usuario y contraseña para el caso el usuario será *aiasseiss* y la contraseña *12345678*

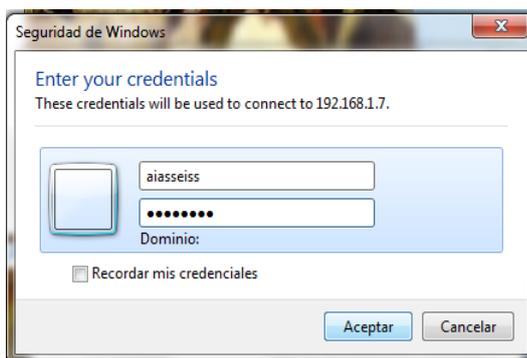


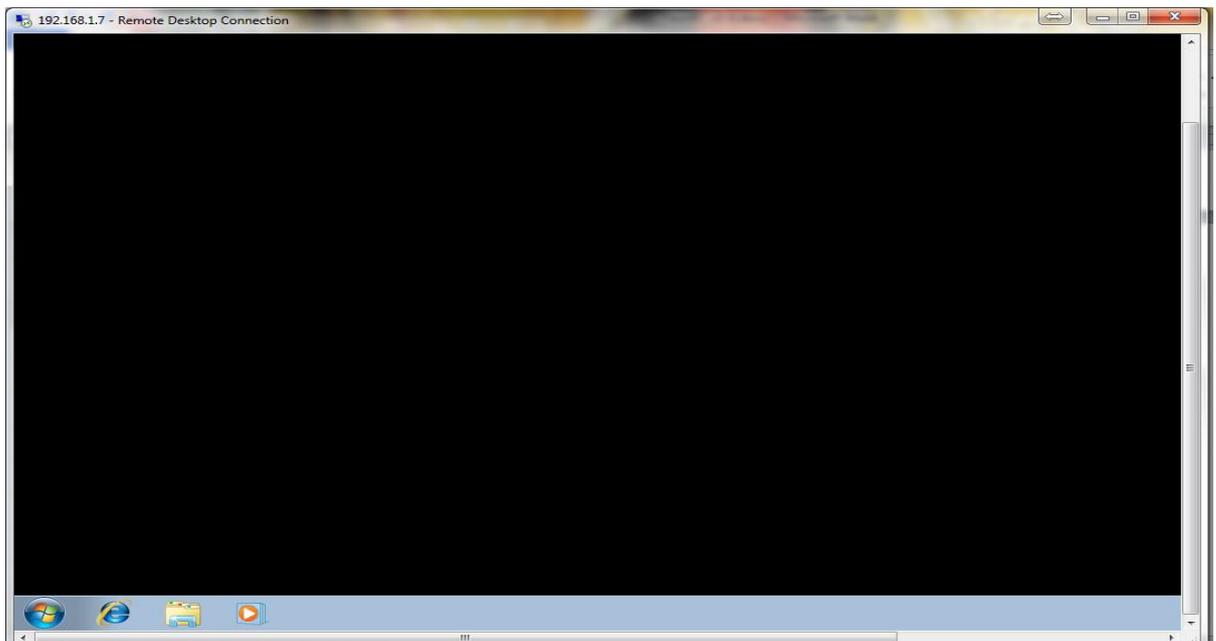
Figura IV – 49. Autenticación VM Window

Posteriormente solicita la verificación de certificados de seguridad en esta pantalla se dará clic en yes.



**Figura IV – 50. Verificación Certificados de Seguridad VM Window**

Realizada las operaciones anteriores podemos ocupar el escritorio remoto de la VM desplegada con la IaaS Eucalyptus como se presenta a continuación en la figura.



**Figura IV – 51. Escritorio Remoto VM Window**

## CONCLUSIONES

Al finalizar el análisis comparativo de las tecnologías IaaS se han obtenido las siguientes conclusiones:

- ✓ Las tres IaaS analizadas presentan características, arquitectura y funcionalidades propias que mediante el adecuado análisis instalación y administración de las IaaS, se puede mejorar notablemente la entrega de recursos computacionales para la EIS.
- ✓ Se fijó un conjunto de parámetros en común y de mayor relevancia entre las IaaS seleccionadas, los mismos que ayudaron a elegir e implementar un prototipo aprovechando todas las características que ofrecen
- ✓ Se aplicó una serie de pruebas a cada una de las IaaS en base a los parámetros establecidos para demostrar la eficiencia de cada una de ellas.
- ✓ En el análisis del parámetro Configuración Nimbus y OpenNebula tienen un porcentaje relativamente alto del 81,25% y 93,75 en relación a Eucalyptus que presenta un porcentaje relativamente menor del 68,75%, brindando varias facilidades para la implementación y administración, entre ellas mínima ejecución de comandos, conocimientos básicos sobre varios aspectos de Linux, cantidad de tiempo y recurso humano relativamente pequeño

- ✓ En el análisis del parámetro Flexibilidad Eucalyptus es superior a las dos IaaS analizadas debido que brinda mayor soporte a hipervisores el uso de DHCP es automático, el despliegue de los servicios es dinámico y presenta variedad de entornos de programación en los que ha sido desarrollado la IaaS.
  
- ✓ La Gestión de Usuarios por parte de las IaaS Nimbus y OpenNebula presentan una interfaz CLI con una amplia gama de comandos propios, en relación con Eucalyptus que presenta su interfaz WEB, misma que provee recursos para la autenticación de usuarios con credenciales de seguridad propias.
  
- ✓ Eucalyptus y OpenNebula presentan el mismo nivel para Gestión de Nodos en comparación con Nimbus que muestra mayor complejidad en el momento de realizar las operaciones necesarias para el funcionamiento y administración del nodo.
  
- ✓ El eje central de las IaaS es despliegue de servicio, Eucalyptus y Nimbus presentan el mismo porcentaje para Gestionar de Máquinas Virtuales, es decir comandos y arquitectura propia. Las tres IaaS en el indicador Variedad de Sistemas Operativos presentan disponibilidad con distribuciones de Sistemas Operativos Linux.
  
- ✓ Se implementó un prototipo de IaaS usando Eucalyptus en base a los resultados del análisis, que permite desplegar y entregar servicios computacionales a usuarios de la EIS.

## RECOMENDACIONES

Al finalizar el trabajo de tesis se han establecido las siguientes recomendaciones:

- ✓ Los parámetros de comparación para la evaluación y análisis comparativo deben ser seleccionados cuidadosamente para analizar las características que permitan la comprobación de hipótesis fácilmente.
- ✓ Analizar detenidamente las funcionalidades, características y arquitectura de las IaaS de manera que permitan medir al máximo sus prestaciones y por ende obtener el resultado deseado en el trabajo de tesis
- ✓ Eucalyptus es una de las mejores opciones al momento de implementar un prototipo IaaS ya que presenta información variada, paquetería disponible y brinda mejores prestaciones en relación con las otras IaaS comparadas.
- ✓ Previa la implementación de la IaaS es importante realizar un estudio de la infraestructura hardware a nivel de servidores con los que cuenta la organización.
- ✓ Para realizar pruebas, experimentos e implementaciones muy interesantes se recomienda trabajar con Centos 5,8 o superior en los Front End, XEN como hypervisor en los Nodos ya que se basan en paquetes bien estables y ajustados por distros de linux.

- ✓ Se recomienda tomar como base investigativa la información obtenida durante el desarrollo de la investigación para implementar niveles superiores de cloud computing con Open Source.

## RESUMEN

Análisis comparativo de Tecnologías de Infraestructura como Servicio en Cloud Computing para Nubes Privadas y su aplicación de un modelo de Infraestructura como Servicio para la Escuela de Ingeniería en Sistemas de la Escuela Superior Politécnica de Chimborazo. De acuerdo al método científico, se desarrolló un prototipo de Infraestructura como Servicio (**IaaS**) usando ambientes de prueba configurados en dos servidores con software Eucalyptus Nimbus y OpenNebula, usando técnicas como revisión bibliográfica para la configuración y administración, observación de ejecución de comandos y comparación de resultados.

Se estableció cinco parámetros de evaluación: configuración, flexibilidad, gestión de usuarios, gestión de nodos y gestión de máquinas virtuales, obteniendo los siguientes puntajes Eucalyptus 85%, Nimbus 77,91% y OpenNebula 82.08%. Se comprobó la hipótesis en base a la observación directa de la relación porcentual de Eucalyptus 29,79%, Nimbus 22,70% y OpenNebula 26.87%

Se concluye que la tecnología IaaS idónea en base al análisis realizado es Eucalyptus, la misma que aporta a mejorar la configuración administración y eficiencia de recursos informáticos disponibles en la Escuela de Ingeniería en Sistemas.

Se recomienda la utilización, configuración y administración de la IaaS Eucalyptus esta permite entregar infraestructura virtual dotando a los usuarios memoria disco duro procesador y sistema operativo de manera ágil y oportuna.

## SUMMARY

Comparative analysis of Technologies Infrastructure as a Service in Cloud Computing for Private Clouds and its application of a model of Infrastructure as a Service for the Systems Engineering School of the Escuela Superior Politécnica de Chimborazo.

According to the scientific method, a prototype of Infrastructure as a Service (IaaS) was developed by using test environments configured in two servers with Eucalyptus Nimbus and OpenNebula software, by using techniques such as literature review for configuration and administration, observation of command execution and comparison results.

Five evaluation parameters were established: configuration, flexibility, user management, node management and virtual machine management, with the following scores Eucalyptus 85%, OpenNebula 77,91% and Nimbus 82.08%. The hypothesis was checked in base on direct observation with a percentage ratio of Eucalyptus 29.79%, OpenNebula 22.70% and Nimbus 26.87%

It is concluded that the ideal IaaS technology based on the analysis performed is Eucalyptus, which provides the same configuration to improve the administration and efficiency of computing resources available of Systems Engineering School.

It is recommended the use, configuration and management of Eucalyptus IaaS infrastructure deliver this allows users endowing virtual memory disk processor and operating system in agile and opportune way.

## GLOSARIO

**Cloud computing:** es un nuevo modelo de prestación de servicios de negocio y tecnología, que permite al usuario acceder a un catálogo de servicios estandarizados y responder con ellos a las necesidades de su negocio, de forma flexible y adaptativa, en caso de demandas no previsibles o de picos de trabajo, pagando únicamente por el consumo efectuado, o incluso gratuitamente en caso de proveedores que se financian por publicidad o de organizaciones sin ánimo de lucro.

**Clusters:** se aplica a los conjuntos o conglomerados de computadoras construidos mediante la utilización de hardware comunes y que se comportan como si fuesen una única computadora

**Grids:** es una tecnología innovadora que permite utilizar de forma coordinada todo tipo de recursos (entre ellos cómputo, almacenamiento y aplicaciones específicas) que no están sujetos a un control centralizado

**Housing:** es una modalidad de alojamiento web destinado principalmente a grandes empresas y a empresas de servicios web

**IaaS (Infrastructure as a Service):** es un modelo de distribución de infraestructura de computación como un servicio, normalmente mediante el uso de una plataforma de virtualización.

**TI (Tecnologías de la Información):** es el estudio, diseño, desarrollo, implementación, soporte o dirección de los sistemas de información computarizados, en particular de software de aplicación y hardware de computadoras. Se ocupa del uso de las computadoras y software para convertir, almacenar, proteger, procesar, transmitir y recuperar la información.

## ANEXOS

## **Anexo 1**

### **Instalación y Administración de Nimbus (7)**

#### **Instalación**

##### **Instalación en el Frond End**

Para la instalación del Frond End se requiere como prerequisite la instalación del SO Centos 5.6 o superior

#### **Instalación de Dependencias**

##### **Dependencias para Python**

Necesitamos varias dependencias para la configuración de Python las que se instalaran desde el repositorio para ello ejecutaremos el siguiente comando

```
# yum install bzip2-devel autoconf db4-devel expat-devel gdbm-devel gmp-devel libGL-  
devel libffi-devel ncurses-devel openssl-devel readline-devel sqlite-devel systemtap-sdt-devel  
tcl-devel tix-devel tk-devel valgrind-devel zlib-devel gcc
```

##### **Instalación de Python 2.5.6**

Para la instalación de Python procedemos a descargar el lenguaje desde la página oficial de Python ejecutando el siguiente comando.

```
# wget http://www.python.org/ftp/python/2.5.6/Python-2.5.6.tgz
```

Descomprimir el archivo descargado mediante la ejecución del siguiente comando

```
# tar xvzf Python-2.5.6.tgz
```

Ingresamos al directorio donde se descomprimió el archivo ejecutando el siguiente comando

```
# cd Python-2.5.6
```

Ejecutamos el siguiente comando para permitir que se configure la preinstalación de Python

```
# ./configure  
# make  
# make install
```

Una vez configurado Python vamos a ejecutar el siguiente comando para verificar la correcta instalación.

```
# python -V  
# python -c "import sqlite3"
```

Verificamos la instalación de paquetes de java que se instalan en el momento de instalar el sistema operativo

```
# java -version  
# javac -version
```

### **Instalación de Apache Ant**

Instalamos desde el repositorio ejecutando el siguiente comando

```
# yum install ant-nodeps
```

Verificamos la correcta instalación de apache ant ejecutando el siguiente comando

```
# ant -version
```

### **Instalación del Servicio Nimbus**

Creamos el usuario unimbus y le asignamos la contraseña ejecutando el siguiente comando

```
# adduser unimbus
```

```
# passwd unimbus
```

Descargamos el archivo de servicio de Nimbus de su página oficial

```
# wget http://www.nimbusproject.org/downloads/nimbus-iaas-2.9-src.tar.gz
```

Creamos un directorio donde se alojará los archivos de configuración del servicio de Nimbus

```
# mkdir -p /opt/nimbus
```

Procedemos a autenticarnos como el usuario creado unimbus y descomprimos el paquete descargado e instalamos el servicio

```
$ tar xvzf nimbus-iaas-2.9-src.tar.gz
$ cd nimbus-iaas-2.9-src
$ ./install /opt/nimbus/
```

Iniciamos el servicio de apache httpd en este caso ejecutamos el comando para permitir que el servicio arranque al iniciar la máquina en el siguiente inicio del SO y lanzamos el servicio.

```
# chkconfig httpd on
# /etc/init.d/httpd start
```

Para iniciar el servicio Nimbus vamos ejecutar el siguiente comando

```
$ /opt/nimbus/bin/nimbusctl start
```

Ejecutado el comando anterior podremos visualizar la siguiente figura

---

```
[unimbus@snimbus ~]$ /opt/nimbus/bin/nimbusctl start
Launching Nimbus services... OK
Launching Cumulus services... OK
```

## Instalación del Cloud Client Nimbus

Añadimos un nuevo usuario en este caso llamado rcsalazarg le asignamos la contraseña necesitada para su autenticación, para ello se ejecuta el siguiente comando.

```
# adduser rcsalazarg  
# passwd rcsalazarg
```

Creamos un directorio temporal para los certificados del nuevo usuario ejecutando el siguiente comando.

```
# mkdir -p /tmp/rcsalazarg
```

Para la creación de certificados del usuario ejecutamos el siguiente comando.

```
[unimbus@snimbus ~]$ /opt/nimbus/bin/nimbus-new-user -d /tmp/rcsalazarg/  
r_salazarg@epoch.edu.ec
```

Descargamos el paquete referente a los clientes de Nimbus de la página oficial ejecutando el siguiente comando

```
$ wget http://www.nimbusproject.org/downloads/nimbus-cloud-client-021.tar.gz
```

Nos autentificamos como el usuario para proceder a configurar el *cloud client nimbus*, descomprimos el paquete descargado ejecutando el siguiente comando

```
$ tar xvzf nimbus-cloud-client-021.tar.gz
```

Copiamos los certificados del directorio temporal a nuestro directorio final y posteriormente cambiamos de dueño a dichos archivos ejecutando los siguientes comandos.

```
# cp /tmp/user/cloud.properties /home/rcsalazarg /nimbus-cloud-client-021/conf/  
# cp /tmp/rcsalazarg/*.pem /home/rcsalazarg/.nimbus/  
# chown rcsalarzag:rcsalazarg /home/rcsalazarg/.nimbus/ -R  
#chown rcsalazarg:rcsalazarg /home/rcsalazarg/nimbus-cloud-client-021/lib/certs/ -R
```

Realizamos una prueba para verificar la correcta configuración del usuario ejecutando el siguiente comando una vez autenticado como dicho usuario y se visualizará los resultados mostrados en rectángulos.

```
$ nimbus-cloud-client-021/bin/cloud-client.sh -list
```

```
No files.
```

```
$ nimbus-cloud-client-021/bin/cloud-client.sh -status
```

```
Querying for ALL instances.
```

```
There's nothing running on this cloud that you own.
```

## Configuración de la Red

Con la ayuda del Editor Vi procedemos a cambiar algunas opciones necesarias para el funcionamiento de la Red en esta IaaS

```
# vi /opt/nimbus/services/etc/nimbus/workspace-service/network-pools/public
```

Añadimos la siguiente línea que nos indica la dirección ip que va a ser asignada al momento de solicitar un servicio dentro de la IaaS.

```
# hostname ipaddress gateway broadcast subnetmask [MAC]  
pub02 172.30.104.253 172.30.104.250 172.30.104.255 255.255.255.0 A2:AA:BB:37:2F:25
```

## Instalación de DHCP

Lo realizamos desde el repositorio ejecutando el siguiente comando

```
# yum install dhcp
```

Copiamos los archivos de ejemplo del Nimbus para modificar con la ayuda del Editor

VI

```
# cp /opt/nimbus/services/var/nimbus/dhcpd.entries /etc/
```

Aquí asignamos la entrada correspondiente para nuestro hostname con la ayuda del

Editor VI

```
# vi /etc/dhcpd.entries
```

Añadimos las siguientes líneas

```
host pub02 {  
    hardware ethernet A2:AA:BB:37:2F:25;  
    fixed-address 172.30.104.253;  
    option host-name "pub02";  
    option subnet-mask 255.255.255.0;  
    option routers 172.30.104.250;  
    option broadcast-address 172.30.104.255;  
    option domain-name-servers 172.30.104.250;  
}
```

Ahora editaremos el archivo de configuración de DHCP

```
# vi /etc/dhcpd.conf
```

Añadiendo las siguiente líneas o simplemente descomentando

```
## GLOBAL OPTS ##  
# Option definitions common or default to all supported networks  
# Keep this:  
ddns-update-style none;  
# Can be overridden in host entry:  
default-lease-time 259200;  
max-lease-time 518400;  
include "/etc/dhcpd.entries";
```

```
## SUBNETS ##  
# Make an entry like this for each supported subnet. Otherwise, the DHCP  
# daemon will not listen for requests on the interface of that subnet.  
subnet 172.30.104.0 netmask 255.255.255.0 {  
}
```

Iniciamos el servicio DHCP ejecutando el siguiente comando

```
# service dhcpd restart
```

## Instalación de Nimbus en el Nodo

### Instalación de XEN

Para instalar Xen desde el repositorio debemos ejecutar el siguiente comando

```
# yum install xen
```

### Configurar el GRUB

Ahora tenemos que configurar el GRUB para que Centos inicie con la versión del Kernel modificado con XEN. Con la ayuda del editor VI modificamos el archivo *menu.lst* que tiene la configuración de boot del sistema mediante la ejecución del siguiente comando

```
# vi /boot/grub/menu.lst
```

Posteriormente cambiaremos el orden de boot

```
default=0  
timeout=5  
splashimage=(hd0,0)/grub/splash.xpm.gz  
hiddenmenu  
title CentOS (2.6.18-308.8.2.el5xen)  
    root (hd0,0)
```

```
kernel /xen.gz-2.6.18-308.8.2.el5
module /vmlinuz-2.6.18-308.8.2.el5xen ro root=LABEL=/ rhgb quiet
module /initrd-2.6.18-308.8.2.el5xen.img
title CentOS (2.6.18-308.8.2.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-308.8.2.el5 ro root=LABEL=/ rhgb quiet
initrd /initrd-2.6.18-308.8.2.el5.img
```

En este caso editamos el default = 0 para que inicie con la versión del kernel propia de XEN y procedemos a reiniciar el servidor ejecutando el siguiente comando.

```
# restart
```

### **Configurar XEN**

Una vez reiniciado el servidor procedemos a configurar el archivo de XEN habilitando opciones con la ayuda del editor VI ejecutando el siguiente comando

```
# vi /etc/xen/xend-config.sxp
```

Aquí cambiamos las siguientes opciones

```
(xend-http-server yes)
(xend-unix-server yes)
```

Reiniciamos el servicio XEN

```
# /etc/init.d/xend restart
```

### **Instalación de Bridge**

Lo realizamos desde el repositorio ejecutando el siguiente comando

```
# yum install bridge-utils
```

Con la ayuda del Editor VI crearemos el puente por el cual existe comunicación de RED ejecutando el siguiente comando

```
# vi /etc/sysconfig/network-scripts/ifcfg-br0
```

Añadiremos las siguientes líneas

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=none
BROADCAST=172.30.104.255
IPADDR=172.30.104.252
NETMASK=255.255.255.0
NETWORK=172.30.104.0
GATEWAY=172.30.104.250
ONBOOT=yes
```

### **Configuración de Libvirt**

Procedemos a crear un usuario, el mismo creado para la administración de Nimbus unimbus ejecutando el siguiente comando

```
# useradd unimbus
# passwd unimbus
```

Creamos un nuevo grupo y asignamos a ese grupo el usuario creado anteriormente ejecutando el siguiente comando

```
# groupadd libvirt
# usermod -a -G libvirt unimbus
```

Con la ayuda del Editor VI cambiamos opciones del archivo de libvirt para permitir que el usuario unimbus ejecute comandos relacionados con libvirt

```
# vi /etc/libvirt/libvirtd.conf
```

Cambiamos opciones de tal manera que subsista de la siguiente manera

```
unix_sock_group = "libvirtd"  
unix_sock_ro_perms = "0770"  
unix_sock_rw_perms = "0770"  
unix_sock_dir = "/var/run/libvirt"  
auth_unix_ro = "none"  
auth_unix_rw = "none"
```

Reiniciamos el servicio libvirtd para que se ejecute los cambios realizados en el archivo de configuración ejecutando el siguiente comando

```
# /etc/init.d/libvirtd restart
```

### **Instalación de workspace control VMM Node**

Para la instalación del nodo procedemos a descargar de la página oficial de Nimbus el workspace control para el Nodo ejecutando el siguiente comando

```
# wget http://www.nimbusproject.org/downloads/nimbus-iaas-controls-2.9.tar.gz
```

Descomprimos el archivo mediante la ejecución del siguiente comando

```
# tar xvzf nimbus-iaas-controls-2.9.tar.gz
```

Creamos el directorio donde se van alojar los archivos del servicio de workspace control ejecutando el siguiente comando

```
# mkdir /opt/nimbus
```

Movemos el contenido del paquete descargado al directorio creado ejecutando el siguiente comando

```
# mv nimbus-iaas-controls-2.9/workspace-control/* /opt/nimbus/
```

Realizamos una prueba de verificación ejecutando el siguiente comando

```
# [ -f /opt/nimbus/bin/workspace-control.sh ] && echo "Correct."
```

```
[root@sxnimbus ~]# [ -f /opt/nimbus/bin/workspace-control.sh ] && echo "Correct."
"
Correct.
```

Modificamos los permisos al contenido del workspace control para que se ejecute correctamente el nodo.

```
# cd /opt/nimbus/
# chown -R root bin etc lib libexec src
# chown -R unimbus var
# find . -type d -exec chmod 775 {} \;
# find . -type f -exec chmod 664 {} \;
# find bin sbin libexec -iname "*sh" -exec chmod 755 {} \;
```

Realizamos una prueba para verificar las dependencias ejecutando el siguiente comando

```
# /opt/nimbus/sbin/test-dependencies.sh
```

```
[root@sxnimbus ~]# /opt/nimbus/sbin/test-dependencies.sh
Python 2.4.3 (#1, Jun 18 2012, 08:55:31) | [GCC 4.1.2 20080704 (Red Hat 4.1.2-5
2)]
OK, looks like the Python dependencies are set up.
```

## Cambio de Parámetros

Con la ayuda del Editor VI cambiamos varios parámetros en diferentes archivos

```
# vi /opt/nimbus/etc/workspace-control/libvirt.conf
```

Añadimos o descomentamos lo siguiente

```
vmm: xen3
```

En el archivo

```
# vi /opt/nimbus/libexec/workspace-control/mount-alter.sh
```

Añadimos o descomentamos las siguientes líneas

```
DRYRUN="true"
```

### **Configuramos el archivo de red de la workspace control**

Usando el editor VI procedemos a cambiar varias opciones de este archivo

```
# vi /opt/nimbus/etc/workspace-control/networks.conf
```

Añadimos o descomentamos las siguientes líneas

```
default: br0  
br0: 172.30.104.0-172.30.104.255  
eatablesconfig: xen-eatables-config.sh  
br0: eth0
```

### **Configuración SUDO**

Para permitir la ejecución de **unimbus** de manera directa a los archivos de la workspace control con la ayuda del Editor VI

```
# vi /etc/sudoers
```

Añadimos las siguientes líneas

```
unimbus ALL=(root) NOPASSWD: /opt/nimbus/libexec/workspace-control/mount-alter.sh  
unimbus ALL=(root) NOPASSWD: /opt/nimbus/libexec/workspace-control/dhcp-config.sh  
unimbus ALL=(root) NOPASSWD: /opt/nimbus/libexec/workspace-control/tmp-lease.sh  
unimbus ALL=(root) NOPASSWD: /opt/nimbus/libexec/workspace-control/xen-eatables-  
config.sh
```

## Configuración de SSH

Ejecutamos los siguientes comandos para generar llaves de autenticación ssh

```
$ ssh-keygen
$ cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
$ scp -r ~/.ssh/ unimbus@sxnimbus:~/
```

## Autoconfiguración del Nodo en Service Nimbus

Ejecutamos el siguiente comando

```
$ /opt/nimbus/bin/nimbus-configure --autoconfig
```

Ahí nos desplegará las siguientes opciones que se presentan a continuación

```
Is the current account (nimbus) the one the service will run under? y/n:
y
Pick a VMM to test with, enter a hostname:
sxnimbus
How much RAM (MB) should be allocated for VMs on the 'sxnimbus' VMM?
1000
Is the current account (nimbus) also the account the privileged scripts will run under
on the VMM (sxnimbus)? y/n:
y
Does the container account (unimbus) need a special (non-default) SSH key to access
the 'nimbus' account on the VMM nodes? y/n:
n
Even if you plan on ever setting up just one VMM and it is localhost to the container,
you should still pick a hostname here ('localhost' if you must)
*** It looks like you have a hostname set up: snimbus
Would you like to manually enter a different hostname? y/n:
```

n

Is your local SSHd server on a port different than 22? Enter 'n' or a port number:

n

If you have not followed the instructions for setting up workspace control yet, please do the basic installation steps now.

Waiting for you to install workspace control for the account 'nimbus' on the test VMM '**sxnimbus**'. After this is accomplished, press return to continue.

**Press enter**

On '**sxnimbus**', did you install workspace-control somewhere else besides '/opt/nimbus/bin/workspace-control.sh'? y/n:

n

We are looking for the directory on the VMM to push customization files from the container node. This defaults to '/opt/nimbus/var/workspace-control/tmp'

Did you install workspace-control under some other base directory besides /opt/nimbus? y/n:

n

So far, no configurations have been changed. The following adjustments will be made based on the questions and tests we just went through:

- The account running the container/service: nimbus
- The hostname running the container/service: service
- The contact address of the container/service for notifications: nimbus@service (port 22)
- The test VMM: vmm1
- The available RAM on that VMM: 1000
- The privileged account on the VMM: unimbus
- The workspace-control path on VMM: /opt/nimbus/bin/workspace-control.sh

- The workspace-control tmpdir on VMM: /opt/nimbus/var/workspace-control/tmp

These settings are now stored in '/home/nimbus/nimbus/services/share/nimbus-autoconfig/autoconfig-decisions.sh'

If you type 'y', that script will be run for you with the settings.

Or you can answer 'n' to the next question and adjust this file.

And then manually run '/home/nimbus/nimbus/services/share/nimbus-autoconfig/autoconfig-adjustments.sh' at your leisure.

OK, point of no return. Proceed? y/n

**y**

\*\*\* Running /home/nimbus/nimbus/services/share/nimbus-autoconfig/autoconfig-adjustments.sh . . .

# Nimbus auto-configuration: make adjustments #

Read settings from '/home/nimbus/nimbus/services/share/nimbus-autoconfig/autoconfig-decisions.sh'

...

Finished.

Realizamos una prueba de verificación para listar que el nodo se encuentra correctamente configurado ejecutando el siguiente comando

```
# /opt/nimbus/bin/nimbus-nodes -l
```

Si la configuración está correctamente realizada se nos presentará la siguiente figura

```
[unimbus@snimbus ~]$ /opt/nimbus/bin/nimbus-nodes -l
hostname      : sxnimbus
pool          : default
memory        : 1000
memory available : 1000
networks      : *
in_use        : false
active        : true
```

## Administración de Nimbus

Transferencia de imágenes de SO a cumulus lo realizamos ejecutando el siguiente comando

```
$ nimbus-cloud-client-021/bin/cloud-client.sh --transfer --sourcefile /tmp/euca-centos-5.3-i386/centos.5-3.x86.img
```

Solicitamos un listado de las imágenes de SO transferidas al cumulus ejecutando el siguiente comando

```
$ nimbus-cloud-client-021/bin/cloud-client.sh -list
```

Se nos presentará un listado similar al mostrado en la siguiente figura

```
[rcsalazarg@snimbus ~]$ nimbus-cloud-client-021/bin/cloud-client.sh -list
[Image] 'centos.5-3.x86.img'          Read/write
      Modified: Aug 21 2012 @ 18:24  Size: 1049624576 bytes (~1001 MB)

[Image] 'nimbus-z2c'                 Read/write
      Modified: Aug 18 2012 @ 11:21  Size: 288358400 bytes (~275 MB)
```

Cambiamos las propiedades para que el arranque del servicio sea de acuerdo a lo requerido por el usuario con la ayuda del editor VI ejecutando el siguiente comando

```
# vi nimbus-cloud-client-021/conf/cloud.properties
```

Aquí modificamos lo siguiente asignando manualmente memoria, cpu necesitados y el lugar de almacenamiento disco duro.

```
vws.memory.request=256
vws.cores.request=1
vws.metadata.mountAs=sda3
```

Iniciamos el servicio ejecutando el siguiente comando

```
$ nimbus-cloud-client-021/bin/cloud-client.sh --run --name centos.5-3.x86.img --hours 1
```

Posteriormente se presenta la siguiente pantalla

```
[rcsalazarg@snimbus ~]$ nimbus-cloud-client-021/bin/cloud-client.sh --run --name
centos.5-3.x86.img --hours 1

Launching workspace.

Workspace Factory Service:
  https://snimbus:8443/wsrf/services/WorkspaceFactoryService

Creating workspace "vm-016"... done.

      IP address: 172.30.104.253
      Hostname: pub02
      Start time: Tue Nov 13 17:20:46 ECT 2012
      Shutdown time: Tue Nov 13 18:20:46 ECT 2012
      Termination time: Tue Nov 13 18:22:46 ECT 2012

Waiting for updates.

"vm-016" reached target state: Running

Running: 'vm-016'
[rcsalazarg@snimbus ~]$
```

Verificaremos el estado de las máquinas que se encuentran ejecutándose desplegando el siguiente comando.

```
# nimbus-cloud-client-021/bin/cloud-client.sh --status
```

Posteriormente se nos presentará la siguiente figura que detalla el correcto despliegue del servicio

```
[rcsalazarg@snimbus ~]$ nimbus-cloud-client-021/bin/cloud-client.sh --status
Querying for ALL instances.

[*] - Workspace #10. 172.30.104.253 [ pub02 ]
      State: Running
      Duration: 60 minutes.
      Start time: Tue Nov 13 17:20:46 ECT 2012
      Shutdown time: Tue Nov 13 18:20:46 ECT 2012
      Termination time: Tue Nov 13 18:22:46 ECT 2012
      *Handle: vm-016
      Image: centos.5-3.x86.img
```

Para la finalización del servicio desplegado ejecutamos el siguiente comando

```
# nimbus-cloud-client-021/bin/cloud-client.sh --terminate --handle vm-016
```

Una vez ejecutado el comando anterior se presenta la siguiente figura

```
[rcsalazarg@snimbus ~]$ nimbus-cloud-client-021/bin/cloud-client.sh --terminate  
--handle vm-016
```

Terminating workspace.

- Workspace handle (EPR): '/home/rcsalazarg/nimbus-cloud-client-021/history/vm-016/vw-epr.xml'

Destroying vm-016... destroyed.

## Anexo 2

### Instalación y Administración de OpenNebula (8)

#### Instalación OpenNebula

#### Instalación en el Frond End

#### Instalación de Dependencias

#### Instalación de GCC

Para instalar el compilador integrado del proyecto GNU lo realizamos desde el repositorio ejecutando el siguiente comando

```
# yum install gcc-c++
```

#### Instalación de Ruby

Instalamos desde el repositorio ejecutado el siguiente comando

```
# yum install ruby
```

#### Instalación de Sqlite

Descargamos desde la página oficial de sqlite ejecutando el siguiente comando

```
# wget http://www.sqlite.org/sqlite-amalgamation-3.6.14.2.tar.gz
```

Descomprimos el paquete descargado ejecutando el siguiente comando

```
# tar -xvzf sqlite-amalgamation-3.6.14.2.tar.gz
```

Ingresamos al directorio creado y ejecutamos los siguientes comandos

```
# cd sqlite-3.6.14.2/  
# ./configure  
# make
```

```
# make install
```

## **Instalación de RUBYGEMS**

Descargamos desde la página oficial de ruby ejecutando el siguiente comando

```
# wget http://production.cf.rubygems.org/rubygems/rubygems-1.8.7.tgz
```

Descomprimos el paquete descargado ejecutando el siguiente comando

```
# tar xvzf rubygems-1.8.7.tgz
```

Ingresamos al directorio creado y ejecutamos los siguientes comandos

```
# cd rubygems-1.8.7
```

```
# ruby setup.rb
```

Instalamos el gem desde repositorio

```
# yum install gem
```

Actualizamos los gem del sistema

```
# gem update --system
```

## **Instalación del YAML**

Para la instalación del formato de serialización de datos legibles por humanos lo descargaremos desde la página oficial

```
# wget http://pyyaml.org/download/libyaml/yaml-0.1.4.tar.gz
```

Descomprimos el paquete descargado ejecutando el siguiente comando

```
# tar xvzf yaml-0.1.4.tar.gz
```

Ingresamos al directorio y ejecutamos los siguientes comandos

```
# cd yaml-0.1.4
# ./configure --prefix=/usr/local
# make
# make install
```

## **Instalación de SCONS**

Descargamos el paquete desde la página oficial de Scons ejecutando el siguiente comando

```
# wget http://prdownloads.sourceforge.net/scons/scons-1.2.0-1.noarch.rpm
```

Instalamos este paquete ejecutando repositorios locales con el siguiente comando

```
# yum localinstall scons-1.2.0-1.noarch.rpm
```

## **Instalación de XMLRPC**

Descargamos los paquetes desde las páginas oficiales y ejecutamos los siguientes comandos

```
# wget http://centos.karan.org/el5/extras/testing/i386/RPMS/xmlrpc-c-1.06.18-1.el5.kb.i386.rpm
# wget http://centos.karan.org/el5/extras/testing/i386/RPMS/xmlrpc-c-devel-1.06.18-1.el5.kb.i386.rpm
```

Realizamos la instalación desde repositorio local ejecutando los siguientes comandos

```
# rpm -i xmlrpc-c-1.06.18-1.el5.kb.i386.rpm
# rpm -i xmlrpc-c-devel-1.06.18-1.el5.kb.i386.rpm
```

## **Instalamos NOKOGIRI**

Para ello ejecutaremos el siguiente comando

```
# gem install nokogiri rake xmlparser
# gem install json
# gem install rack
# gem install sinatra
# gem install thin
```

## Instalación de OpenNebula

Creamos el usuarios que va a ser el administrador de la IaaS ejecutando el siguiente comando

```
#adduser oneadmin
#passwd oneadmin
```

Creamos el directorio donde va ser repositorio de ficheros de servicio y configuración de OpenNebula

```
# mkdir /srv/cloud/one
```

Descargamos desde la página oficial de OpenNebula

```
#wget -P /tmp http://dev.opennebula.org/attachments/download/264/opennebula-2.0.1.tar.gz
```

Ingresamos al directorio temporal donde descargamos el paquete de OpenNebula y ejecutamos los siguientes comandos

```
# pushd /tmp
# tar xvzf opennebula-2.0.1.tar.gz
# pushd opennebula-2.0.1
# scons
#./install.sh -u oneadmin -g oneadmin -d /srv/cloud/one
# popd
# popd
```

Con el editor VI editamos opciones de alias en el fichero .bashrc

```
# vi .bashrc
```

Añadimos las siguientes líneas

```
# User specific aliases and functions
export ONE_LOCATION=/srv/cloud/one
export ONE_AUTH=$HOME/.one_auth
export ONE_XMLRPC=http://localhost:2633/RPC2
export PATH=/srv/cloud/one/bin:$PATH
```

Compilaremos el fichero o archivo ejecutando el siguiente comando

```
# source ~/.bashrc
```

Añadimos el fichero de autenticación ejecutando el siguiente comando

```
# vi .one_auth
```

Ingresamos dentro del archivo las siguientes líneas que detallan el usuario y contraseña para la inicio del servicio OpenNebula

```
oneadmin:123456
```

Iniciación del Servicio ejecutamos el siguiente comando autenticado con el usuario oneadmin.

```
# one start
```

Después de la ejecución del comando mencionado se nos presentará la siguiente imagen

```
[oneadmin@sonebu ~]$ one start
oned and scheduler started
```

## Configuración del Driver

Con la ayuda del Editor VI vamos a modificar el fichero de configuración de OpenNebula para ello ejecutaremos el siguiente comando

```
vi /srv/cloud/one/etc/oned.conf
```

Añadir o descomentar las siguientes líneas

```
# XEN Information Driver Manager Configuration
#-----
IM_MAD = [
    name      = "im_xen",
    executable = "one_im_ssh",
    arguments = "xen" ]
XEN Virtualization Driver Manager Configuration
#-----
VM_MAD = [
    name      = "vmm_xen",
    executable = "one_vmm_ssh",
    arguments = "xen",
    default   = "vmm_ssh/vmm_ssh_xen.conf",
    type      = "xen" ]
# SSH Transfer Manager Driver Configuration
#-----
TM_MAD = [
    name      = "tm_ssh",
    executable = "one_tm",
    arguments = "tm_ssh/tm_ssh.conf" ]
```

## Instalación de Nimbus en el Nodo

### Instalación de XEN

Para instalar Xen desde del repositorio debemos ejecutar el siguiente comando

```
# yum install xen
```

## Configurar el GRUB

Ahora tenemos que configurar el GRUB para que Centos inicie con la versión del Kernel modificado con XEN. Con la ayuda del editor VI modificamos el archivo menu.lst que tiene la configuración de boot del sistema mediante la ejecución del siguiente comando

```
# vi /boot/grub/menú.lst
```

Posteriormente cambiaremos el orden de boot

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-308.8.2.el5xen)
    root (hd0,0)
    kernel /xen.gz-2.6.18-308.8.2.el5
    module /vmlinuz-2.6.18-308.8.2.el5xen ro root=LABEL=/ rhgb quiet
    module /initrd-2.6.18-308.8.2.el5xen.img
title CentOS (2.6.18-308.8.2.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-308.8.2.el5 ro root=LABEL=/ rhgb quiet
initrd /initrd-2.6.18-308.8.2.el5.img
```

En este caso editamos el default = 0 para que inicie con la versión del kernel propia de XEN y procedemos a reiniciar el servidor ejecutando el siguiente comando.

```
# restart
```

## Configurar XEN

Una vez reiniciado el servidor procedemos a configurar el archivo de XEN habilitando opciones con la ayuda del editor VI ejecutando el siguiente comando

```
# vi /etc/xen/xend-config.sxp
```

Aquí cambiamos las siguientes opciones

```
(xend-http-server yes)
(xend-unix-server yes)
```

Reiniciamos el servicio XEN

```
# /etc/init.d/xend restart
```

Creamos el usuario que va a ser gestor de autenticación entre el nodo y el front end de la IaaS ejecutando el siguiente comando

```
# useradd -d /srv/cloud/one/ -s /bin/bash oneadmin
#passwd oneadmin
```

## Configuración de SSH

Ejecutamos los siguientes comandos para generar llaves de autenticación ssh

```
$ ssh-keygen
$ cp ~/.ssh/id_rsa.pub ~/.ssh/authorized_keys
$ scp -r ~/.ssh/ oneadmin@ sxonebu:~/
```

## Instalación de Dependencias

El nodo requiere de varias dependencias las cuales las instalaremos desde el repositorio ejecutando el siguiente comando.

```
# yum install foo glibc glibc-common glibc-devel cpp glibc-headers kernel-headers libgomp
libstdc++-devel nscd gcc-c++ rpm-build yum-utils pkgconfig libxml2 libxml2-devel expat
expat-devel libxslt libxslt-devel openssl openssl-devel curl curl-devel
```

## Instalación de Ruby

La instalación de ruby se lo realizará desde el repositorio ejecutando el siguiente comando

```
# yum -y install ruby ruby-libs ruby-devel ruby-irb ruby-docs ruby-rdoc ruby-ri rubygems
cmake
```

## Administración de OpenNebula

### Creación de Usuarios

Para la creación de usuarios ejecutaremos el siguiente comando

```
$ oneuser create rcsalazarg 123456
```

### Listar Usuarios

Para listar usuarios ejecutaremos el siguiente comando

```
$ oneuser list
```

Posteriormente se desplegará la siguiente figura

```
[oneadmin@sonebu ~]$ oneuser list
  ID USER                                PASSWORD
  ---
  0 oneadmin                              7c4a8d09ca3762af61e59520943dc26494f8941b
  1 rcsalazarg                             7c4a8d09ca3762af61e59520943dc26494f8941b
```

### Eliminar usuario

Para eliminar un usuario ejecutamos el siguiente comando

```
$ oneuser delete 1
```

## Gestión de Nodos

### Creación de Nodos

Para la creación de un nodo se ejecutará el siguiente comando

```
$ onehost create sxonebu im_xen vmm_xen tm_ssh
```

### Listar Nodos

Para listar nodos ejecutaremos el siguiente comando

```
$ onehost list
```

Después se presenta la siguiente figura en la que se muestra información referente a aspectos técnicos del nodo como cantidad de memoria porcentaje de CPU y el estado en se encuentra el nodo

```
[oneadmin@sonebu ~]$ onehost list
```

ID	NAME	CLUSTER	RVM	TCPU	FCPU	ACPU	TMEM	FMEM	STAT
0	sxonebu	default	0	100	97	100	1023M	124M	on

```
[oneadmin@sonebu ~]$
```

### Monitoreo de Nodos

Para el monitoreo del nodo se ejecutará el siguiente comando

```
$ onehost show 0
```

Posteriormente se despliega la siguiente figura

```
[oneadmin@sonebu ~]$ onehost show 0
```

```
HOST 0 INFORMATION
```

---

```
ID : 0
NAME : sxonebu
CLUSTER : default
STATE : MONITORED
IM_MAD : im_xen
VM_MAD : vmm_xen
TM_MAD : tm_ssh

HOST SHARES
MAX MEM : 1047552
USED MEM (REAL) : 920576
USED MEM (ALLOCATED) : 0
MAX CPU : 100
USED CPU (REAL) : 1
USED CPU (ALLOCATED) : 0
RUNNING VMS : 0

MONITORING INFORMATION
ARCH=1686
CPUSPEED=2665
FREECPU=99
FREEMEMORY=126976
HOSTNAME=sxonebu.esepoch.edu.ec
HYPERVISOR=xen
MODELNAME=Intel(R) Core(TM) i5 CPU M 480 @ 2.67GHz
NETRX=262
NETTX=5569
TOTALCPU=100
TOTALMEMORY=1047552
USEDCPU=1
USEDMEMORY=920576
```

```
[oneadmin@sonebu ~]$
```

El monitoreo con actualización continua lo realizaremos ejecutando el siguiente comando

```
$ onehost top
```

### **Eliminar un nodo**

Para eliminar un nodo ejecutaremos el siguiente comando

```
$ onehost delete 0
```

### **Activar nodo**

Para activar un nodo ejecutaremos el siguiente comando

```
$ onehost enable 0
```

### **Desactivar nodo**

Para desactivar un nodo ejecutamos el siguiente comando

```
$ onehost disable 0
```

## **Gestión de Máquinas Virtuales**

### **Creación de Red**

Para el proceso de creación de la red, con la ayuda del Editor VI vamos a crear el fichero de la red ejecutando el siguiente comando

```
$ vi redlan
```

En el fichero ingresaremos las siguientes líneas que detallan información sobre la red a usarse por la máquina virtual

```
NAME = "Red LAN"
```

```
TYPE = RANGED
# This vnet can be only used by the owner user
PUBLIC = NO
#Now we'll use the cluster private network (physical)
BRIDGE = br0
NETWORK_SIZE = C
NETWORK_ADDRESS = 172.30.104.0
#Custom Attributes to be used in Context
GATEWAY = 172.30.104.250
DNS = 192.168.1.1
LOAD_BALANCER = 172.30.104.252
```

Una vez creado el archivo con opciones de la Red procedemos a ejecutar el siguiente comando

```
# onevnet -v create redlan
```

## Mostrar la RED

Para mostrar información de la Red instaurada vamos ejecutar el siguiente comando

```
$ onevnet show 2
```

Posteriormente se presentará la siguiente pantalla

```
[oneadmin@sonebu ~]$ onevnet show 2
VIRTUAL NETWORK 2 INFORMATION
-----
ID:          : 2
UID:         : 0
PUBLIC      : N

VIRTUAL NETWORK TEMPLATE
BRIDGE=br0
DNS=192.168.1.1
GATEWAY=172.30.104.250
LOAD_BALANCER=172.30.104.252
NAME=Red LAN
NETWORK_ADDRESS=172.30.104.0
NETWORK_SIZE=C
TYPE=RANGED

LEASES INFORMATION
LEASE=[ IP=172.30.104.230, MAC=02:00:ac:1e:68:e6, USED=1, VID=24 ]
[oneadmin@sonebu ~]$
```

## Publicación de la Red

Para la publicación de la red ejecutaremos el siguiente comando

```
$ onevnet publish 2
```

## Eliminar Red

Para eliminar la red ejecutaremos el siguiente comando

```
$ onevnet delete 2
```

## Listar la Red

Para listar las redes disponibles y publicadas ejecutaremos el siguiente comando

```
$ onevnet list
```

Posteriormente se nos presentará la siguiente figura

```
[oneadmin@sonebu ~]$ onevnet list
```

ID	USER	NAME	TYPE	BRIDGE	P	#LEASES
2	oneadmin	Red LAN	Ranged	br0	Y	1

## Registro de Imágenes

Para el registro de imágenes usaremos el Editor VI para crear el fichero con información referente a recursos que la imagen necesita para su registro y publicación

```
$ vi imagencentos
```

Añadiremos las siguientes líneas

```
NAME = "CentOS 5.3"  
PATH = /home/oneadmin/euca-centos-5.3-i386/centos.5-3.x86.img  
PUBLIC = YES  
DESCRIPTION = "Centos 5.5 x86_64 para pruebas del proyecto  
Formiga CLOUD"
```

Para el registro de la imagen procedemos a ejecutar el siguiente comando

```
$ oneimage register imagencentos
```

## Listar Imágenes Registradas

Para listar las imágenes registradas ejecutaremos el siguiente comando

```
$ oneimage list
```

Posteriormente se presentará la siguiente figura que indica las el listado de imágenes registradas

```
[oneadmin@sonebu ~]$ oneimage list
```

ID	USER	NAME	TYPE	REGTIME	PUB	PER	STAT	#VMS
0	oneadmin	CentOS 5.3	OS	Sep 07, 2012 18:50	Yes	No	used	3

## Mostrar Imágenes Registradas

Para mostrar imágenes ejecutaremos el siguiente comando

```
$ oneimage show 0
```

Se presentará la siguiente figura que contiene información referente al identificador, si es imagen pública o persistente, ubicación del archivo imagen, estado entre otros.

```
[oneadmin@sonebu ~]$ oneimage show 0
```

---

IMAGE	INFORMATION
ID	: 0
NAME	: CentOS 5.3
TYPE	: OS
REGISTER TIME	: 09/07 13:50:11
PUBLIC	: Yes
PERSISTENT	: No
SOURCE	: /srv/cloud/one/var//images/b817207d88ed793e32f80b847b2472ae93f4
deb5	
STATE	: used
RUNNING_VMS	: 3

IMAGE TEMPLATE
DESCRIPTION=Centos 5.5 x86_64 para pruebas del proyecto
Formiga CLOUD
DEV_PREFIX=hd
NAME=CentOS 5.3
PATH=/home/oneadmin/euca-centos-5.3-1386/centos.5-3.x86.img

## Publicación de Imagen

Para la publicación de la imagen ejecutamos el siguiente comando

```
$ oneimage publish 0
```

## Activar Imágenes

Para activar una imagen ejecutaremos el siguiente comando

```
$ oneimage enable 0
```

## Creación de Máquinas Virtuales

Para la creación de máquinas virtuales usaremos el Editor VI para crear el fichero con información necesaria para desplegar una máquina virtual ejecutaremos el siguiente comando.

```
$ vi creatmv
```

Añadimos las siguientes líneas al fichero, las que describen opciones de memoria, cpu, disco duro a ser usado por la máquina virtual.

```
REQUIREMENTS = "HYPERVISOR=\"xen\""  
NAME = CentOSMV  
CPU = 1  
MEMORY = 256  
DISK = [ image = "CentOS 5.3" ]  
DISK = [ type = swap, size = 1024 ]  
NIC = [ NETWORK = "Red LAN", IP = "172.30.104.253" ]  
OS = [ bootloader = "/usr/bin/pygrub" ]  
RAW = [ type="xen", data="on_crash=destroy" ]
```

Una vez creado el fichero con la información necesaria para el uso de recursos por la máquina virtual ejecutaremos el siguiente comando que nos permite crear una máquina virtual.

```
$ onevm create crearmv
```

## Listar las Máquinas Virtuales

Para listar máquinas virtuales ejecutaremos el siguiente comando

```
$ onevm list
```

```
[oneadmin@sonebu ~]$ onevm list
```

ID	USER	NAME	STAT	CPU	MEM	HOSTNAME	TIME
1	oneadmin	Centos		0	OK	sxonebu	00 00:00:29
5	oneadmin	centos5-		0	OK	sxonebu	00 00:00:27
10	oneadmin	CentOSMV		0	OK	sxonebu	00 00:00:43
24	oneadmin	ttylinux		0	OK	sxonebu	00 00:00:13

```
[oneadmin@sonebu ~]$
```

## Monitoreo de Máquinas Virtuales

Para monitorear una máquina virtual ejecutaremos el siguiente comando

```
$ onevnet show 1
```

Posteriormente nos desplegará la siguiente figura en la que se detalla información concerniente a todos los recursos que usa la máquina virtual para iniciar su servicio

```
VIRTUAL MACHINE 1 INFORMATION
-----
ID          : 1
NAME       : Centos
STATE      :
LCM_STATE  : LCM_INIT
START TIME : 09/11 15:16:41
END TIME   : 09/11 15:17:10
DEPLOY ID: : -

VIRTUAL MACHINE MONITORING
USED MEMORY : 0
USED CPU    : 0
NET_TX      : 0
NET_RX      : 0

VIRTUAL MACHINE TEMPLATE
CPU=0.1
DISK=[
  CLONE=YES,
  DISK_ID=0,
  IMAGE=CentOS 5.3,
  IMAGE_ID=0,
  READONLY=NO,
  SAVE=NO,
  SOURCE=/srv/cloud/one/var//images/b817207d88ed793e32f80b847b2472ae93f4deb5,
  TARGET=sda3,
  TYPE=DISK ]
FEATURES=[
  ACPI=no ]
MEMORY=256
NAME=Centos
NIC=[
  BRIDGE=eth1,
  IP=172.30.104.249,
  MAC=02:00:ac:1e:68:f9,
  NETWORK=Red LAN,
  NETWORK_ID=1 ]
VMID=1
```

Para monitorear de manera automática debemos ejecutar el siguiente comando

```
$ onevm top
```

Después de la ejecución de este comando se presentará la siguiente figura

ID	USER	NAME	STAT	CPU	MEM	HOSTNAME	TIME
1	oneadmin	Centos		0	OK	sxonebu	00 00:00:29
5	oneadmin	centos5-		0	OK	sxonebu	00 00:00:27
10	oneadmin	CentOSMV		0	OK	sxonebu	00 00:00:43
24	oneadmin	ttylinux		0	OK	sxonebu	00 00:00:13

### Despliegue de Máquinas Virtuales

Para el despliegue de máquinas virtuales ejecutaremos el siguiente comando el primer identificador es de la máquina virtual el segundo del nodo donde deseamos que se ejecute

```
$ onevm deploy 1 0
```

### Suspensión de Máquinas Virtuales

Para suspender una máquina virtual ejecutaremos el siguiente comando

```
$ onevm suspend 1
```

### Apagar Máquinas Virtuales

Para la apagar de la máquina virtual ejecutaremos el siguiente comando

```
$ onevm shutdown 1
```

### Reiniciar la Máquina Virtual

Para la reiniciar la máquina virtual ejecutaremos el siguiente comando

```
$ onevm restart 1
```

## **Eliminar Máquina Virtual**

Para la eliminar la máquina virtual ejecutaremos el siguiente comando

```
$ onevm delete 1
```

## **BIBLIOGRAFÍA**

### **1. CLOUD COMPUTING**

[http://es.wikipedia.org/wiki/Computaci%C3%B3n\\_en\\_la\\_nube](http://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube)

2012/01/30

### **2. EUCALYPTUS**

<http://www.ibm.com/developerworks/ssa/linux/library/1-kvm-virtual-server/index.html>

2012/07/10

### **3. EVOLUCIÓN DEL CLOUD COMPUTING**

[http://www.google.com.ec/url?sa=t&rct=j&q=cloud%20computing%20retos%20y%20oportunidades&source=web&cd=2&cad=rja&ved=0CCMQFjAB&url=http%3A%2F%2Fwww.ontsi.red.es%2Fonsi%2Fsites%2Fdefault%2Ffiles%2F1-estudio\\_cloud\\_computing\\_retos\\_y\\_oportunidades\\_vdef.pdf&ei=WzOlUP\\_4K4yo8QSih4CwBw&usg=AFQjCNGLWyGOHG9rENgkrufK69W7dDcvmw](http://www.google.com.ec/url?sa=t&rct=j&q=cloud%20computing%20retos%20y%20oportunidades&source=web&cd=2&cad=rja&ved=0CCMQFjAB&url=http%3A%2F%2Fwww.ontsi.red.es%2Fonsi%2Fsites%2Fdefault%2Ffiles%2F1-estudio_cloud_computing_retos_y_oportunidades_vdef.pdf&ei=WzOlUP_4K4yo8QSih4CwBw&usg=AFQjCNGLWyGOHG9rENgkrufK69W7dDcvmw)

2012/05/20

### **4. HYPERVISOR**

<http://es.wikipedia.org/wiki/Hipervisor>

2012/05/30

## **5. INFRAESTRUCTURA COMO SERVICIO**

<http://www.technoreeze.com/2011/07/15/cloud-computing-v-infraestructura-como-servicio-iaas/>

2012/02/10

## **6. INSTALACIÓN Y ADMINISTRACIÓN DE EUCALYPTUS**

[http://www.google.com.ec/url?sa=t&rct=j&q=instalacion%20de%20eucalyptus&source=web&cd=2&ved=0CDIQFjAB&url=https%3A%2F%2Fwww.cesga.es%2Fgl%2Fbiblioteca%2FdownloadAsset%2Fid%2F603&ei=AbGIUMLSqbp0gHbp4CQAg&usg=AFQjCNEE3KLkhszyEL1f65NWBUU4L\\_Wp0A&cad=rja](http://www.google.com.ec/url?sa=t&rct=j&q=instalacion%20de%20eucalyptus&source=web&cd=2&ved=0CDIQFjAB&url=https%3A%2F%2Fwww.cesga.es%2Fgl%2Fbiblioteca%2FdownloadAsset%2Fid%2F603&ei=AbGIUMLSqbp0gHbp4CQAg&usg=AFQjCNEE3KLkhszyEL1f65NWBUU4L_Wp0A&cad=rja)

2012/09/10

## **7. INSTALACIÓN Y ADMINISTRACIÓN DE NIMBUS**

[http://www.google.com.ec/url?sa=t&rct=j&q=nimbus\\_installation.pdf&source=web&cd=2&cad=rja&ved=0CDUQFjAB&url=http%3A%2F%2Fpercomp.nchc.org.tw%2Fdata%2Fuploaded%2Ffile%2FNimbus\\_Installation.pdf&ei=6LGIUKS8EMeQ0QH9hoHYAQ&usg=AFQjCNHkyLpPuhT3bS8iDuFg5Rg4cKRIVg](http://www.google.com.ec/url?sa=t&rct=j&q=nimbus_installation.pdf&source=web&cd=2&cad=rja&ved=0CDUQFjAB&url=http%3A%2F%2Fpercomp.nchc.org.tw%2Fdata%2Fuploaded%2Ffile%2FNimbus_Installation.pdf&ei=6LGIUKS8EMeQ0QH9hoHYAQ&usg=AFQjCNHkyLpPuhT3bS8iDuFg5Rg4cKRIVg)

2012/09/20

## **8. INSTALACIÓN Y ADMINISTRACIÓN DE OPENNEBULA**

<http://www.google.com.ec/url?sa=t&rct=j&q=instalacion%20de%20opennebula&source=web&cd=1&cad=rja&ved=0CCsQFjAA&url=https%3A%2F%2Fwww.cesga.es%2Fgl%2Fbiblioteca%2FdownloadAsset%2Fid%2F602&ei=T7KIUL3>

[XKrOI0QGM34DQAw&usg=AFQjCNGaE\\_bXkzeOPgc3\\_US1yELN9a74Eg](http://XKrOI0QGM34DQAw&usg=AFQjCNGaE_bXkzeOPgc3_US1yELN9a74Eg)

2012/10/10

## **9. INSTALACIÓN Y ADMINISTRACIÓN DE XEN**

[http://tuxjm.net/docs/Administracion de Servidores Virtuales con Xen y GNU Linux/html-multiples/instalando-xen-en-centos-5.html](http://tuxjm.net/docs/Administracion_de_Servidores_Virtuales_con_Xen_y_GNU_Linux/html-multiples/instalando-xen-en-centos-5.html)

2012/09/12

## **10. KVM**

<http://www.ibm.com/developerworks/ssa/linux/library/l-kvm-virtual-server/index.html>

2012/06/06

## **11. NIMBUS**

<http://www.nimbusproject.org/docs/2.10/>

2012/07/15

## **12. OPENNEBULA**

<http://opennebula.org/>

2012/08/20

### **13. TIPOS DE HYPERVISORES**

<http://blog.art4software.com/2012/05/virtualizacion-i-introduccion-hypervisor/>

2012/06/01

### **14. VENTAJAS DE LA IAAS**

<http://blog.webstudio.es/archives/444>

2012/02/11

### **15. XEN**

[http://wiki.xen.org/wiki/Xen\\_Overview/es](http://wiki.xen.org/wiki/Xen_Overview/es)

2012/06/05