

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

IMPLEMENTACIÓN DE UN PROTOTIPO DE *CLOUD COMPUTING* DE MODELO PRIVADO PARA OFRECER INFRAESTRUCTURA COMO SERVICIO

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y REDES DE INFORMACIÓN**

ANA CRISTINA GUERRERO ALEMÁN

anac.guerreroa@gmail.com

ELISA KARINA MENA MALDONADO

elikary86@gmail.com

DIRECTOR: IVÁN BERNAL CARRILLO, Ph. D.

imbernal@mailfie.epn.edu.ec

Quito, febrero 2011

DECLARACIÓN

Nosotras, Elisa Karina Mena Maldonado y Ana Cristina Guerrero Alemán, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Elisa Karina Mena Maldonado

Ana Cristina Guerrero Alemán

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Elisa Karina Mena Maldonado y Ana Cristina Guerrero Alemán, bajo mi supervisión.

Iván Bernal Carrillo, Ph. D.

DIRECTOR DE PROYECTO

AGRADECIMIENTOS

A Diosito y la Virgencita de la Merced por haberme dado la oportunidad de venir a este mundo pero sobretodo por haberme regalado una familia maravillosa.

A mis padres por haberme enseñado a ser una mujer de bien. Mami gracias por el ejemplo de profesional, de madre, de esposa y de amiga que me has dado. Papi gracias por tu paciencia tus consejos y tu apoyo incondicional. A los dos, gracias, muchas gracias por ser mis padres, los amo con todo mi corazón.

A mi hermano Santiago por su amistad y sabios consejos durante mi vida estudiantil y profesional, por haberme incentivado a ser responsable, honesta y por siempre creer en mí. A Mirian por su amistad, a Francisco y Gabriel por mostrarme el lado tierno de la vida.

A Daniel Carrasco por su asesoría profesional, por su gran cariño, apoyo, comprensión y por hacer de cada día un día feliz.

Al Doctor Iván Bernal, excelente director, por su orientación, dedicación, disponibilidad, confianza y ayuda durante el tiempo que duró este proyecto.

A Ana Cris por haber sido una gran compañera y amiga.

A la Ingeniera Lourdes Custodio, Alejandra Villenas, Karla Cedeño, a las personas de la UGI Cristian Loza y Gabriel López y a todos quienes de alguna forma contribuyeron con su apoyo moral y humano necesarios para la realización de este proyecto.

Muchísimas gracias a todos...

Elisa Karina

AGRADECIMIENTOS

Son tantas las personas que me han apoyado durante el transcurso de mis estudios profesionales, que es muy difícil mencionar a cada una de ellas; sin embargo, quiero que tengan presente que sus palabras de aliento, buenos deseos y motivación, siempre las llevaré conmigo.

Agradezco a Dios por darme su bendición, a mi familia, en especial a mis padres Hugo y Mariela por darme las fuerzas para día a día seguir adelante, por enseñarme que los obstáculos hay que aprender a enfrentarlos y sobre todo a superarlos, gracias por su apoyo constante e incondicional.

A mi hermano Víctor por todo el apoyo brindado en cada momento que lo he necesitado y a mi hermano Marco por su cariño y confianza.

Muchas gracias a mis maestros, a cada uno de los cuales les tengo una gran admiración y respeto; y quienes durante estos años han inculcado en mí responsabilidad, dedicación y perseverancia. Un infinito gracias al Doctor Iván Bernal por ser nuestro mentor y con su disponibilidad, orientación y confianza permitir que este proyecto salga adelante.

A mi amiga y compañera Elisa por su tiempo, dedicación y paciencia.

Finalmente, y aunque no puedo incluir todos los nombres, agradezco a mis amigos y compañeros con quienes he tenido la oportunidad de compartir momentos buenos y malos, disfrutando siempre de su apoyo y amistad.

Ana Cristina

DEDICATORIA

Dedicado a mis padres. Este trabajo simboliza la infinita gratitud y admiración que siento por ustedes.

Feliz trigésimo aniversario.

Elisa Karina

Dedico este trabajo a mis padres quienes me han brindado la invaluable oportunidad de culminar mi carrera. A mi hermano Víctor quien ha sido mi más grande ejemplo profesional y la persona que me ha brindado el empuje necesario para poder cumplir todas mis metas.

Ana Cristina

CONTENIDO

CONTENIDO.....	I
ÍNDICE DE FIGURAS	V
ÍNDICE DE TABLAS	IX
ÍNDICE DE CÓDIGOS	X
RESUMEN	XI
PRESENTACIÓN.....	XV
CAPÍTULO 1.....	1
1.1 INTRODUCCIÓN.....	2
1.2 DEFINICIÓN DE <i>CLOUD COMPUTING</i>	4
1.3 TAXONOMÍA DE <i>CLOUD COMPUTING</i>	6
1.3.1 CONSUMIDOR DEL SERVICIO (<i>SERVICE CONSUMER</i>).....	7
1.3.2 PROVEEDOR DEL SERVICIO (<i>SERVICE PROVIDER</i>).....	8
1.3.3 DESARROLLADOR DEL SERVICIO (<i>SERVICE DEVELOPER</i>)	10
1.4 MODELOS DE INFRAESTRUCTURA EN <i>CLOUD COMPUTING</i>	10
1.4.1 <i>CLOUD COMPUTING</i> DE MODELO PÚBLICO.....	10
1.4.2 <i>CLOUD COMPUTING</i> DE MODELO PRIVADO	13
1.4.3 <i>CLOUD COMPUTING</i> DE MODELO HÍBRIDO	14
1.5 ARQUITECTURA EN CAPAS Y MODELOS DE SERVICIO DE <i>CLOUD COMPUTING</i>	16
1.5.1 HARDWARE	18
1.5.2 VIRTUALIZACIÓN	18
1.5.2.1 Virtualización y <i>Cloud Computing</i>	19
1.5.3 SERVICIOS.....	20
1.5.3.1 Infraestructura como Servicio (IaaS).....	20
1.5.3.2 Plataforma como Servicio (PaaS)	22
1.5.3.3 Software como Servicio (SaaS).....	24
1.5.4 CARACTERÍSTICAS COMUNES DE LOS TRES SERVICIOS DE <i>CLOUD COMPUTING</i>	25
1.6 VENTAJAS Y DESVENTAJAS DE <i>CLOUD COMPUTING</i>	29
1.7 RELACIÓN CON OTRAS TECNOLOGÍAS	34
1.7.1 <i>GRID COMPUTING</i>	34
1.7.2 WEB 2.0	35
1.7.3 <i>UTILITY COMPUTING</i>	36

CAPÍTULO 2.....	38
2.1 UBUNTU SERVER EDITION-KARMIC KOALA (9.10)	39
2.1.1 UBUNTU ENTERPRISE CLOUD (UEC).....	40
2.2 EUCALYPTUS.....	40
2.2.1 ARQUITECTURA DE EUCALYPTUS	42
2.2.1.1 Controlador del Nodo (<i>Node Controller, NC</i>) [42].....	44
2.2.1.2 Controlador del <i>Cluster</i> (<i>Cluster Controller, CC</i>).....	46
2.2.1.3 Controlador de la Nube (<i>Cloud Controller, CLC</i>).....	47
2.2.1.4 <i>Walrus</i>	48
2.2.1.5 Controlador de Almacenamiento (<i>Storage Controller, SC</i>)	48
2.2.2 ADMINISTRACIÓN, ACCESO Y RED VIRTUAL DE EUCALYPTUS	49
2.2.2.1 Interfaz del Cliente (<i>Client Interface</i>)	50
2.2.2.2 Interfaz de Administración (<i>Administrative Interface</i>).....	50
2.2.2.3 Administración de zonas de disponibilidad (<i>availability zones</i>).....	51
2.2.2.4 Red Virtual de <i>Eucalyptus</i>	52
2.3 AMAZON WEB SERVICES (AWS)	56
2.4 OTRAS OPCIONES DE SISTEMA OPERATIVO Y DE MIDDLEWARE PARA CLOUD COMPUTING.....	56
2.5 INSTALACIÓN Y CONFIGURACIÓN DE LOS COMPONENTES DE UBUNTU ENTERPRISE CLOUD57	
2.5.1 REQUERIMIENTOS DE HARDWARE	58
2.5.1.1 <i>Front – End</i>	58
2.5.1.2 Nodo.....	59
2.5.2 DISEÑO DE LA TOPOLOGÍA DE RED Y UBICACIÓN DE LOS COMPONENTES FÍSICOS60	
2.5.3 INSTALACIÓN DE UBUNTU ENTERPRISE CLOUD 9.10.....	63
2.5.3.1 <i>Front-End</i>	63
2.5.3.2 Nodo.....	68
2.5.4 CONFIGURACIÓN DE LOS COMPONENTES DE EUCALYPTUS.....	69
2.5.4.1 Herramientas de línea de comandos (<i>euca2ools</i>).....	76
2.5.4.2 Archivos de configuración de <i>fron-end</i> y nodos.....	78
2.5.4.3 Modos de Red de <i>Eucalyptus</i>	83
2.5.4.3.1 Modo System	83
2.5.4.3.2 Modo Static	83
2.5.4.3.3 Modo Managed.....	84
2.5.4.3.4 Modo Managed-NoVLAN	84

2.5.5	PROCESO DE INSERCIÓN DE IMÁGENES EN <i>EUCALYPTUS</i>	85
2.5.6	EJECUCIÓN DE INSTANCIAS (MÁQUINAS VIRTUALES)	93
2.5.7	GRUPOS DE SEGURIDAD Y <i>ELASTIC IP ADDRESSES</i> EN <i>EUCALYPTUS</i>	97
2.5.7.1	<i>Elastic IP addresses</i>	97
2.5.7.2	Grupos de seguridad	98
2.5.8	INSTALACIÓN Y CONFIGURACIÓN DE <i>ELASTICFOX</i>	100
CAPÍTULO 3.....		106
3.1	CUENTAS DE USUARIO	107
3.2	UTILIZACIÓN DE HERRAMIENTAS DE GESTIÓN A NIVEL DE USUARIO	111
3.2.1	INSTALACIÓN DE <i>EUCA2TOOLS</i>	111
3.2.2	UTILIZACIÓN DE <i>ELASTICFOX</i> PARA CONEXIÓN DE SSH CON PUTTY Y CREACIÓN DE LLAVES (<i>KEYPAIRS</i>)	114
3.3	ADMINISTRACIÓN DE VOLÚMENES	119
3.4	IMPLEMENTACIÓN DE UN SERVICIO EN UNA INSTANCIA.....	122
3.4.1	MÉTODO DE ENLACE DE ARCHIVOS	122
3.4.2	MÉTODO <i>CHROOT</i>	128
3.5	PRESTACIONES PARA LOS USUARIOS.....	130
3.6	VELOCIDAD DE TRANSMISIÓN Y TRÁFICO DEL PROTOTIPO.....	132
3.7	USO DE CPU EN LOS NODOS QUE FORMAN PARTE DEL PROTOTIPO	140
CAPÍTULO 4.....		151
4.1	CONCLUSIONES	152
4.2	RECOMENDACIONES	156
BIBLIOGRAFÍA		160
ANEXOS		172
ANEXO A: PROVEEDORES DE <i>CLOUD COMPUTING</i>		173
ANEXO B: VIRTUALIZACIÓN.....		180
B.1	VIRTUALIZACIÓN DE LA PLATAFORMA.....	180
B.1.1	VIRTUALIZACIÓN DE APLICACIONES.....	181
B.1.2	VIRTUALIZACIÓN DE ESCRITORIO.....	181
B.2	VIRTUALIZACIÓN DE EQUIPOS Y SERVIDORES	182
B.2.1	VIRTUALIZACIÓN COMPLETA	182
B.2.2	PARAVIRTUALIZACIÓN.....	183
B.2.3	VIRTUALIZACIÓN HÍBRIDA.....	184
B.3	EXTENSIÓN DE VIRTUALIZACIÓN.....	184

B.3.1	TECNOLOGÍA DE VIRTUALIZACIÓN INTEL (INTEL-VT).....	184
B.3.2	VIRTUALIZACIÓN AMD (AMD-V)	185
B.4	<i>HYPERVISORS</i>	186
B.4.1	XEN	186
B.4.2	KVM	187
B.4.3	QEMU	188
B.4.4	VMWARE	189
	ANEXO C: DUAL CORE	191
	ANEXO D: <i>AMAZON WEB SERVICES (AWS)</i>	192
D.1	<i>EC2TOOLS</i>	194
	ANEXO E: CONFIGURACIÓN DE LA BIOS EN LOS NODOS	196
	ANEXO F: COSTOS DE EQUIPOS UTILIZADOS EN LA IMPLEMENTACIÓN DEL PROTOTIPO.....	198
	ANEXO G: HERRAMIENTA DE GESTIÓN <i>RIGHTSCALE</i>	200
	ANEXO H: HERRAMIENTA DE MONITOREO MRTG	203
	ANEXO I: MANUAL DE CONFIGURACIÓN PARA USUARIOS NO ADMINISTRADORES.....	207

ÍNDICE DE FIGURAS

CAPÍTULO 1

FIGURA 1.1 CONCEPTOS DE <i>CLOUD COMPUTING</i>	6
FIGURA 1.2 TAXONOMÍA PARA <i>CLOUD COMPUTING</i>	7
FIGURA 1.3 <i>CLOUD COMPUTING</i> DE MODELO PÚBLICO.....	11
FIGURA 1.4 <i>CLOUD COMPUTING</i> DE MODELO PRIVADO	13
FIGURA 1.5 <i>CLOUD COMPUTING</i> DE MODELO HÍBRIDO	15
FIGURA 1.6 COMPONENTES Y CAPAS DE <i>CLOUD COMPUTING</i>	17
FIGURA 1.7 ARQUITECTURA EN CAPAS Y MODELO DE SERVICIOS EN <i>CLOUD COMPUTING</i>	17
FIGURA 1.8 INFRAESTRUCTURA COMO SERVICIO.....	21
FIGURA 1.9 PLATAFORMA COMO SERVICIO.....	23
FIGURA 1.10 SOFTWARE COMO SERVICIO	24
FIGURA 1.11 RELACIÓN ENTRE LOS SERVICIOS, APLICACIONES Y TIPOS DE NUBES.....	26
FIGURA 1.12 ARQUITECTURA DE <i>CLOUD COMPUTING</i>	27
FIGURA 1.13 RELACIÓN ENTRE USUARIOS Y PROVEEDORES DE <i>CLOUD COMPUTING</i>	37

CAPÍTULO 2

FIGURA 2.1 LOGOTIPO DE UBUNTU	39
FIGURA 2.2 LOGOTIPO DE UBUNTU 9.10- KARMIC KOALA	39
FIGURA 2.3 LOGOTIPO DE <i>EUCALYPTUS SYSTEMS</i>	41
FIGURA 2.4 CONFIGURACIÓN DE LOS COMPONENTES DE <i>EUCALYPTUS</i>	42
FIGURA 2.5 EJEMPLOS DE CONFIGURACIONES DE <i>EUCALYPTUS</i> EN MÁQUINAS FÍSICAS	43
FIGURA 2.6 VISIÓN GENERAL DE LOS SERVICIOS DEL <i>CLOUD CONTROLLER</i>	47
FIGURA 2.7 REPRESENTACIÓN CONCEPTUAL DE LA NUBE DE <i>EUCALYPTUS</i>	49
FIGURA 2.8 INTERFACES <i>PUBLIC</i> Y <i>PRIVATE</i> DE UNA MV DE <i>EUCALYPTUS</i>	53
FIGURA 2.9 AISLAMIENTO DEL TRÁFICO DE DOS MVS A TRAVÉS DE ETIQUETAS VLAN	55
FIGURA 2.10 LOGO DE <i>AMAZON WEB SERVICES</i>	56
FIGURA 2.11 CONFIGURACIÓN DE LOS EQUIPOS FÍSICOS.....	60
FIGURA 2.12 COMPONENTES DE <i>EUCALYPTUS</i> INSTALADOS EN CADA EQUIPO FÍSICO	61
FIGURA 2.13 DIAGRAMA TOPOLÓGICO DE LAS REDES INVOLUCRADAS EN LA NUBE IMPLEMENTADA....	62
FIGURA 2.14 INSTALACIÓN DE <i>UBUNTU ENTERPRISE CLOUD</i>	64
FIGURA 2.15 CONFIGURACIÓN DEL <i>HOSTNAME</i> DE LA MÁQUINA	64
FIGURA 2.16 MODO DE INSTALACIÓN DE UEC.....	65
FIGURA 2.17 CREACIÓN DE UNA NUEVA CUENTA DE USUARIO	66
FIGURA 2.18 CONFIGURACIÓN DEL SERVIDOR DE CORREO ELECTRÓNICO (<i>POSTFIX</i>).....	66
FIGURA 2.19 CONFIGURACIÓN DEL FQDN PARA EL CORREO ELECTRÓNICO	67
FIGURA 2.20 CONFIGURACIÓN DEL NOMBRE DEL <i>CLUSTER</i> DENTRO DE LA NUBE	67
FIGURA 2.21 CONFIGURACIÓN DE UN CONJUNTO DE IPS “PÚBLICAS”	68
FIGURA 2.22 ADVERTENCIA DEL NO USO DE HARDWARE CON EXTENSIÓN DE VIRTUALIZACIÓN	69
FIGURA 2.23 SERVICIOS DE <i>EUCALYPTUS</i> ACTIVOS EN EL <i>FRONT-END</i>	70
FIGURA 2.24 SERVICIOS DE <i>EUCALYPTUS</i> ACTIVOS EN EL NODO 1	70
FIGURA 2.25 ASIGNACIÓN DE DIRECCIONES IP A CADA INTERFAZ EN EL <i>FRONT-END</i>	72

FIGURA 2.26 ASIGNACIÓN DE DIRECCIONES IP A CADA INTERFAZ EN EL NODO1	73
FIGURA 2.27 DESCARGA DE CREDENCIALES DESDE INTERFAZ WEB.....	75
FIGURA 2.28 AVAILABILITY ZONES EN SMARTCLOUD	77
FIGURA 2.29 IMÁGENES DISPONIBLES EN EL STORE DE UEC	86
FIGURA 2.30 EUCA-BUNDLE-IMAGE (KERNEL)	87
FIGURA 2.31 PARTES DE LA IMAGEN EN /TMP	88
FIGURA 2.32 EUCA-UPLOAD-BUNDLE (KERNEL).....	88
FIGURA 2.33 EUCA-REGISTER (KERNEL)	88
FIGURA 2.34 EUCA-BUNDLE-IMAGE (RAMDISK)	89
FIGURA 2.35 EUCA-UPLOAD-BUNDLE (RAMDISK).....	89
FIGURA 2.36 EUCA-REGISTER (RAMDISK).....	89
FIGURA 2.37 EUCA-BUNDLE-IMAGE ASOCIACIÓN (KERNEL / RAMDISK)	91
FIGURA 2.38 EUCA-UPLOAD-BUNDLE ASOCIACIÓN (KERNEL / RAMDISK).....	92
FIGURA 2.39 EUCA-REGISTER IMAGEN ASOCIACIÓN (KERNEL / RAMDISK).....	92
FIGURA 2.40 IMÁGENES DISPONIBLES EN EL SISTEMA	93
FIGURA 2.41 TIPOS DE MÁQUINA VIRTUAL	94
FIGURA 2.42 MÁQUINA VIRTUAL DE CENTOS EJECUTÁNDOSE.....	95
FIGURA 2.43 IPS “PÚBLICAS” DISPONIBLES.....	95
FIGURA 2.44 ARCHIVO DE ASIGNACIÓN DE DIRECCIONES PRIVADAS POR DHCP	96
FIGURA 2.45 CICLO DE VIDA DE UNA MÁQUINA VIRTUAL.....	97
FIGURA 2.46 EUCA-ALLOCATE-ADDRESS.....	98
FIGURA 2.47 EUCA-ASSOCIATE-ADDRESS	98
FIGURA 2.48 EUCA-ADD-GROUP	99
FIGURA 2.49 EUCA-AUTHORIZE.....	99
FIGURA 2.50 AMAZON WEB SERVICES, ELASTICFOX	101
FIGURA 2.51 ARCHIVO ELASTICFOX.XPI	101
FIGURA 2.52 ELASTICFOX	102
FIGURA 2.53 CONFIGURACIÓN DE CREDENCIALES EN ELASTICFOX	102
FIGURA 2.54 CONFIGURACIÓN DE REGIONS EN ELASTICFOX	103
FIGURA 2.55 CONFIGURACIÓN DE REGIONS EN ELASTICFOX	103
FIGURA 2.56 IMÁGENES VISTAS DESDE ELASTICFOX.....	104

CAPÍTULO 3

FIGURA 3.1 VENTANA DE AUTENTICACIÓN DE USUARIOS.....	107
FIGURA 3.2 FORMULARIO DE APLICACIÓN PARA UNA NUEVA CUENTA.....	108
FIGURA 3.3 APROBACIÓN DE UNA CUENTA DE USUARIO A TRAVÉS DE CORREO ELECTRÓNICO.....	109
FIGURA 3.4 INGRESO A LA CUENTA DEL USUARIO 1 EN UEC	110
FIGURA 3.5 CUENTAS DE USUARIO ACTIVAS.....	111
FIGURA 3.6 INSTALACIÓN EUCA2TOOLS NIVEL USUARIO	112
FIGURA 3.7 ARCHIVO EUCARC USUARIO1.....	112
FIGURA 3.8 AVAILABILITY ZONES CON CREDENCIALES DE USUARIO 1	113
FIGURA 3.9 IMÁGENES DISPONIBLES, VISTAS CON CREDENCIALES DEL USUARIO1	113
FIGURA 3.10 ACCESO A UNA INSTANCIA EN EJECUCIÓN CON LLAVE PRIVADA DE USUARIO1 UTILIZANDO EUCA2TOOLS.....	114
FIGURA 3.11 CREACIÓN DE KEYPAIRS DESDE ELASTICFOX.....	115
FIGURA 3.12 KEYPAIRS DEL USUARIO1 VISTAS DESDE ELASTICFOX.....	115
FIGURA 3.13 INSTANCIA EJECUTÁNDOSE CON CREDENCIALES DEL USUARIO1.....	116

FIGURA 3.14 CONFIGURACIÓN PARA <i>PUTTY</i> EN <i>ELASTICFOX</i>	117
FIGURA 3.15 BOTÓN PARA LA CONEXIÓN A UNA INSTANCIA DESDE <i>ELASTICFOX</i>	118
FIGURA 3.16 INICIO DE SESIÓN EN UNA INSTANCIA DESDE <i>ELASTICFOX</i> A TRAVÉS DE <i>PUTTY</i>	118
FIGURA 3.17 CREACIÓN DE UN VOLUMEN EN <i>ELASTICFOX</i>	120
FIGURA 3.18 ASOCIACIÓN DE UN <i>VOLUMEN</i> A UNA INSTANCIA <i>ELASTICFOX</i>	120
FIGURA 3.19 VOLUMEN ASOCIADO A UNA INSTANCIA <i>ELASTICFOX</i>	121
FIGURA 3.20 VOLUMEN ASOCIADO A UNA INSTANCIA <i>EUCA2TOOLS</i>	121
FIGURA 3.21 SITIO WEB PUBLICADO EN VOL1	124
FIGURA 3.22 BASE DE DATOS “CLOUD” CREADA	126
FIGURA 3.23 VISUALIZACIÓN EN <i>ELASTICFOX</i> DEL <i>SNAPSHOT</i> CREADO	126
FIGURA 3.24 NUEVO VOLUMEN CREADO A PARTIR DE UN <i>SNAPSHOT</i>	127
FIGURA 3.25 VOLUMEN RECONOCIDO COMO DISPOSITIVO EN EL SISTEMA OPERATIVO.....	128
FIGURA 3.26 SITIO WEB PUBLICADO EN UN VOLUMEN LÓGICO CON <i>CHROOT</i>	130
FIGURA 3.27 RESULTADO DE <i>IPTRAF</i> EN ETH1 (8 MVS EN EJECUCIÓN)	133
FIGURA 3.28 RESULTADOS DE <i>IPTRAF</i> EN LA INTERFAZ ETH1 CON LA CONEXIÓN DE USUARIOS A TODAS LAS MÁQUINAS VIRTUALES EN EJECUCIÓN	134
FIGURA 3.29 MÁQUINAS VIRTUALES CON DIRECCIONES “PÚBLICAS”	136
FIGURA 3.30 RESULTADO DE <i>IFTOP</i> : TRÁFICO DE SSH Y WWW A TRAVÉS DE ETH1	137
FIGURA 3.31 RESULTADO DE <i>IFTOP</i> : MUESTRA LA CONEXIÓN ESTABLECIDA ENTRE <i>ELASTICFOX</i> Y LA NUBE EN EL PUERTO 8773.....	137
FIGURA 3.32 RESULTADO DE <i>IFTOP</i> EN ETH0 DEL NODO1 QUE MUESTRA EL TRÁFICO ENTRE EL <i>CLUSTER</i> <i>CONTROLLER</i> Y EL <i>NODE CONTROLLER</i> EN EL PUERTO 8775.....	138
FIGURA 3.33 RESULTADO DE <i>IPTRAF</i> QUE MUESTRA LA VELOCIDAD DE TRANSMISIÓN EN LA INTERFAZ ETH0.....	139
FIGURA 3.34 LISTADO DE MVS EJECUTÁNDOSE EN EL NODO 2	141
FIGURA 3.35 USO DE CPU CUANDO 4 MVS C1.MEDIUM ARRANCAN EN EL NODO 2	141
FIGURA 3.36 USO DE CPU CUANDO LAS 4 MVS C1.MEDIUM ACTUALIZAN SUS REPOSITORIOS	142
FIGURA 3.37 USO DE CPU CUANDO 4 MVS M1.SMALL SE ESTÁN EJECUTANDO Y ACTUALIZAN SUS REPOSITORIOS	143
FIGURA 3.38 VISUALIZACIÓN CON <i>MPSTAT</i> DEL USO DE CPU CUANDO 4 MVS M1.SMALL ACTUALIZAN SUS REPOSITORIOS	143
FIGURA 3.39 LISTADO DE MVS EJECUTÁNDOSE EN EL NODO 1	144
FIGURA 3.40 USO DE CPU CUANDO 8 MVS C1.MEDIUM ARRANCAN EN EL NODO 1	144
FIGURA 3.41 USO DE CPU CUANDO 8 MVS C1.MEDIUM SE ESTÁN EJECUTANDO Y ACTUALIZAN SUS REPOSITORIOS	145
FIGURA 3.42 USO DE CPU CUANDO UNA MV C1.XLARGE SE EJECUTA EN EL NODO1	146
FIGURA 3.43 USO DE CPU CUANDO UNA SEGUNDA MV C1.XLARGE SE ESTÁ EJECUTANDO EN EL NODO 1	147
FIGURA 3.44 USO DE CPU CUANDO 1 MV C1.XLARGE SE EJECUTA EN EL NODO 2.....	148
FIGURA 3.45 USO DE CPU CUANDO 16 MVS C1.MEDIUM SE EJECUTAN EN EL NODO 1	149
FIGURA 3.46 USO DE CPU CUANDO 4 MVS C1.MEDIUM SE EJECUTAN EN EL NODO 2	149

ANEXOS

FIGURA B.1 <i>HYPERVERSOR DEL TIPO NON HOSTED</i>	180
FIGURA B.2 <i>HYPERVERSOR DEL TIPO HOSTED</i>	181
FIGURA B.3 VIRTUALIZACIÓN COMPLETA VS PARAVIRTUALIZACIÓN	182

FIGURA D.1 AWS <i>MANAGEMENT CONSOLE</i>	192
FIGURA E.1 ACTIVACIÓN DE EXTENSIÓN DE VIRTUALIZACIÓN EN LA BIOS.....	196
FIGURA E.2 CARACTERÍSTICAS PROCESADOR INTEL Q8400.....	197
FIGURA G.1 ENLACE A <i>RIGHTSCALE</i> DESDE UEC.....	200
FIGURA G.2 PÁGINA DE INGRESO A <i>RIGHTSCALE</i>	201
FIGURA G.3 REGISTRO DE LA NUBE CON <i>RIGHTSCALE</i>	201
FIGURA G.4 NUBE DE <i>EUCALYPTUS</i> REGISTRADA CON <i>RIGHTSCALE</i>	202
FIGURA H. 1 TRÁFICO EN ETH0 CADA 5 MINUTOS.....	204
FIGURA H.2 TRÁFICO EN ETH0 CADA 30 MINUTOS.....	205
FIGURA H.3 TRÁFICO EN ETH1 CADA 5 MINUTOS.....	205
FIGURA H.4 TRÁFICO EN ETH1 CADA 30 MINUTOS.....	205
FIGURA I.1 INGRESO A LA NUBE SMARTCLOUD.....	207
FIGURA I.2 APLICACIÓN A UNA NUEVA CUENTA.....	207
FIGURA I.3 APLICACIÓN A UNA NUEVA CUENTA.....	208
FIGURA I.4 CUENTA DE CORREO ELECTRONICO DEL NUEVO USUARIO.....	208
FIGURA I.5 ENLACE DE ACTIVACIÓN DE CUENTA.....	209
FIGURA I.6 CUENTA DEL NUEVO USUARIO EN UEC.....	209
FIGURA I.7 HYBRIDFOX EN FIREFOX.....	210
FIGURA I.8 REGIÓN SMARTCLOUD.....	210
FIGURA I.9 CONFIGURACIÓN DE CREDENCIALES.....	211
FIGURA I.10 CREACIÓN DE <i>KEYPAIRS</i>	211
FIGURA I.11 LLAVE PRIVADA <i>KEYPAIRS</i>	212
FIGURA I.12 EJECUTAR UNA INSTANCIA.....	212
FIGURA I.13 EJECUCIÓN DE UNA MÁQUINA VIRTUAL.....	213
FIGURA I.14 MÁQUINA VIRTUAL EJECUTÁNDOSE.....	214
FIGURA I.15 CONFIGURACIÓN DEL CLIENTE SSH.....	214
FIGURA I.16 TRANSFORMACIÓN .PEM A .PPK.....	215
FIGURA I.17 SECURITY GROUP DEFAULT AFRODITA.....	216
FIGURA I.18 LLAVE .PPK.....	216
FIGURA I.19 SESIÓN EN UNA INSTANCIA.....	217
FIGURA I.20 CREACIÓN DE UN VOLUMEN LÓGICO.....	217
FIGURA I.21 ADJUNTAR UN VOLUMEN A UNA MÁQUINA VIRTUAL.....	218
FIGURA I. 22 VOLUMEN RECONOCIDO COMO UN DISPOSITIVO.....	218
FIGURA I.23 FORMATO A UN VOLUMEN LÓGICO.....	219
FIGURA I.24 MONTAJE DEL VOLUMEN LÓGICO.....	219
FIGURA I. 25 CREAR UN SNAPSHOT DE UN VOLUMEN.....	220
FIGURA I.26 VOLUMEN DESDE UN <i>SNAPSHOT</i>	220

ÍNDICE DE TABLAS

CAPÍTULO 1

TABLA 1.1 IMPLEMENTACIÓN TRADICIONAL VS IMPLEMENTACIÓN EN <i>CLOUD COMPUTING</i>	3
TABLA 1.2 OFERTA DE SERVICIOS BAJO DEMANDA	28
TABLA 1.3 VENTAJAS DE <i>CLOUD COMPUTING</i> MODELO PÚBLICO Y MODELO PRIVADO	32
TABLA 1.4 DESVENTAJAS DE <i>CLOUD COMPUTING</i> MODELO PÚBLICO Y MODELO PRIVADO	33
TABLA 1.5 <i>GRID</i> VS. <i>CLOUD COMPUTING</i>	34

CAPÍTULO 2

TABLA 2.1 REQUERIMIENTOS DE HARDWARE PARA UN <i>FRONT-END</i> UEC	58
TABLA 2.2 REQUERIMIENTOS DE HARDWARE PARA UN NODO UEC	59
TABLA 2.3 DIRECCIONES IP ASIGNADAS A CADA MÁQUINA DENTRO DEL SISTEMA	61

CAPÍTULO 3

TABLA 3.1 CÁLCULO DEL MÁXIMO DE USUARIOS CON LA CONFIGURACIÓN ESTABLECIDA.....	131
--	-----

ANEXOS

TABLA A.1 <i>CLOUD COMPUTING INFRASTRUCTURE</i>	174
TABLA A.2 <i>CLOUD COMPUTING PASS AND SAAS PROVIDER</i>	177
TABLA A.3 <i>OPEN SOURCE BASED CLOUD COMPUTING SERVICES</i>	179

ÍNDICE DE CÓDIGOS

CAPÍTULO 2

CÓDIGO 2.1 COPIA DE SSH-ID EN EL NODO 1.....	70
CÓDIGO 2.2 ARCHIVO DE CONFIGURACIÓN DE LAS INTERFACES DE RED DEL <i>FRONT-END</i>	71
CÓDIGO 2.3 ARCHIVO DE CONFIGURACIÓN DE LAS INTERFACES DE RED DEL NODO1	71
CÓDIGO 2.4 REGISTRO DE LOS COMPONENTES DE <i>EUCALYPTUS</i>	73
CÓDIGO 2.5 SCRIPT CREDENTIALSCRIPT.SH	74
CÓDIGO 2.6 ARCHIVO EUCARC ADMINISTRADOR.....	76
CÓDIGO 2.7 ARCHIVO DE CONFIGURACIÓN DE <i>EUCALYPTUS</i> EN EL <i>FRONT-END</i>	80
CÓDIGO 2.8 ARCHIVO DE CONFIGURACIÓN DE <i>EUCALYPTUS</i> EN EL NODO 1.....	81
CÓDIGO 2.9 IMAGEN CENTOS (<i>KERNEL</i>)	87
CÓDIGO 2.10 IMAGEN CENTOS: INSERCIÓN DE LA IMAGEN (<i>RAMDISK</i>)	89
CÓDIGO 2.11 IMAGEN CENTOS: ASOCIACIÓN <i>KERNEL/RAMDISK</i>	90
CÓDIGO 2.12 CREACIÓN DE LAS LLAVES <i>KEYPAIRS</i>	94
CÓDIGO 2.13 EJECUCIÓN DE UNA MÁQUINA VIRTUAL.....	94

CAPÍTULO 3

CÓDIGO 3.1 CREACIÓN DE UN VOLUMEN LÍNEA DE COMANDOS	119
CÓDIGO 3.2 ASOCIACIÓN DE UN VOLUMEN A UNA INSTANCIA CON LA LÍNEA DE COMANDOS.....	121
CÓDIGO 3.3 CONFIGURACIÓN DEL SERVICIO PARA QUE SE INICIE CUANDO LA INSTANCIA ARRANCA	123
CÓDIGO 3.4 CONFIGURACIÓN DE UN VOLUMEN PARA EL SERVIDOR WEB.....	123
CÓDIGO 3.5 INSTALACIÓN DEL SERVIDOR DE BASE DE DATOS <i>MYSQL</i>	125
CÓDIGO 3.6 MONTAR LOS ARCHIVOS QUE <i>MYSQL</i> REQUIERE MANEJAR EN UNA INSTANCIA	125
CÓDIGO 3.7 CREACIÓN DE UN <i>SNAPSHOT</i> UTILIZANDO <i>EUCA2TOOLS</i>	126
CÓDIGO 3.8 CAMBIO DE PROPIETARIO Y GRUPO AL QUE PERTENECE <i>MYSQL</i>	127
CÓDIGO 3.9 INSTALACIÓN DE UN NUEVO SISTEMA OPERATIVO BÁSICO EN UN VOLUMEN LÓGICO	129
CÓDIGO 3.10 INSTALACIÓN DEL SERVIDOR WEB EN EL VOLUMEN LÓGICO	129

ANEXOS

CÓDIGO H.1 ARCHIVO DE CONFIGURACIÓN <i>SNMPD.CONF</i>	203
---	-----

RESUMEN

El adelanto de la ciencia y la innovación de las Tecnologías de Información (TI) son grandes fuerzas motrices para el desarrollo de una serie de actividades sobre todo de carácter económico y accesibilidad de recursos para pequeñas, medianas y grandes empresas y para usuarios en general.

Poco a poco la demanda de servicios relacionados con las nuevas tecnologías va en aumento; el usuario siempre espera un servicio de vanguardia y que satisfaga sus necesidades, es por eso que se van haciendo más comunes aquellos servicios que son medibles y facturables. Para poder implementar este tipo de servicios, aprovechando al máximo los recursos disponibles, ha nacido el concepto de virtualización.

La virtualización es parte importante del desarrollo de las TI, ayuda a los usuarios a satisfacer sus necesidades, de tal forma que ofrece la independencia del hardware, lo que permite el acceso a recursos de forma virtualizada.

Cloud Computing es una de las innovaciones que puede ser adoptada en cualquier momento ya que ofrece entornos virtualizados en una infraestructura automatizada y escalable, para brindar a los usuarios servicios de TI accesibles desde una red privada o una red pública como lo es el Internet, valiéndose de Servicios Web.

Para el usuario, el lugar en donde se encuentre implementado *Cloud Computing* no es importante, ya que para él esto es transparente, sólo le interesa el tipo y la calidad de servicio que pueda manejar.

Muchas veces, los usuarios se ven obligados a utilizar nuevas opciones tecnológicas debido a la presión económica existente, por lo que se ven en la necesidad de reducir costos. En este tipo de casos *Cloud Computing* es una buena solución dentro del campo de la tecnología y que ayuda a reducir costos

de equipos, gracias a su escalabilidad y uso compartido de recursos por parte de múltiples usuarios.

Los servicios que ofrece *Cloud Computing* dependen del modelo de infraestructura que maneje el proveedor, el modelo de infraestructura híbrido es la combinación de los modelos público y privado. En *Cloud Computing* de modelo público, el proveedor ofrece un determinado servicio a través del Internet utilizando todos sus recursos disponibles, y permite el acceso a múltiples usuarios (personas u organizaciones) quienes pagan por utilizar ese servicio bajo demanda. En *Cloud Computing* de modelo privado, la compañía es dueña de los recursos físicos, tiene el control total de las aplicaciones y, puede ofrecer el servicio a sus empleados a través de una red privada.

Cloud Computing pone a disponibilidad del usuario cualquier tipo de recursos como servicios, éstos se han asociado en tres grandes grupos denominados modelos de servicios. Infraestructura como Servicio (IaaS) ofrece como servicio a los clientes varios recursos computacionales, entre los más destacados: abastecimiento de procesamiento, espacio de almacenamiento y equipos de red. Se brinda este servicio mediante una plataforma de virtualización, el usuario no administra la infraestructura de *Cloud Computing* ni incurre en gastos de operación, él sólo tiene el control sobre los sistemas operativos y aplicaciones desplegadas.

Plataforma como Servicio (PaaS) brinda a los usuarios la facilidad de desplegar sus aplicaciones sobre la infraestructura de *Cloud Computing*, basadas en lenguajes y herramientas de programación que el proveedor soporte.

Software como Servicio (SaaS) permite al usuario utilizar cualquier tipo de aplicaciones manejadas por el proveedor, sin la necesidad de que el usuario adquiera, instale o maneje cualquier tipo de actualización del servicio.

Probablemente, sin haberse dado cuenta, muchas personas se han estado deslizando lentamente en un entorno de *Cloud Computing*. Por ejemplo, si se tiene una cuenta de correo basada sólo en la web, ya se es parte del

movimiento *Cloud Computing*. La mayoría de los usuarios de *Yahoo! Mail*, de *Gmail* o de *Hotmail* recuperan, leen y escriben su correo electrónico a través del Internet. Adicionalmente, hay que nombrar algunos ejemplos de *Cloud Computing* como son: *Apple's iTunes store*, *eBay*, *Amazon.com (AWS)*, *Google Apps.*, entre otros que ofrecen sus aplicaciones a través del Internet. Se debe recalcar que *Cloud Computing* es una tecnología que actualmente continúa en desarrollo, pero sin lugar a duda ofrece servicios muy valiosos, que permiten a los usuarios hacer un mejor uso de cualquier tipo de recursos.

En el futuro, *Cloud Computing* va a ir adquiriendo mayor realce e importancia dentro del desarrollo efectivo de un número más grande de empresas que deseen almacenar sus datos y servicios de forma remota, sin importar el equipo terminal que se utilice, el tipo de software que en éste se maneje ni la infraestructura que está detrás de esta tecnología.

Por otro lado, el *middleware Eucalyptus* permite la implementación de modelos de *Cloud Computing* privados e híbridos, para ofrecer Infraestructura como Servicio. Trabaja con varias distribuciones de Linux y tecnologías de virtualización y ofrece una interfaz de acceso compatible con *Amazon Web Services (AWS)*.

La compañía *Amazon* ofrece servicios de *Cloud Computing* a través del Internet denominados *Amazon Web Services (AWS)*, esta colección de Servicios Web va desde el almacenamiento hasta plataformas de bases de datos.

El presente proyecto presenta la implementación de un prototipo de *Cloud Computing* privado, utilizando *Eucalyptus* que forma parte del conjunto de aplicaciones de *Ubuntu Enterprise Cloud (UEC)*. Básicamente, el prototipo ofrece Infraestructura como Servicio (IaaS), es decir, proporciona recursos de hardware de forma virtual a través del despliegue de máquinas virtuales que el usuario puede utilizar para el desarrollo de otros servicios. El cliente para utilizar el servicio que ofrece el prototipo, primero debe registrarse y luego autenticarse en la interfaz web que muestra el prototipo al utilizar *Eucalyptus*.

Fundamentalmente se utilizan tres computadores físicos correspondientes a un *front-end* y dos nodos, éstos manejan componentes de *Eucalyptus* como *Cloud/Cluster/Storage/Walrus Controller (front-end)*, y *Node Controller* (nodos).

El prototipo permite utilizar imágenes de sistemas operativos de software libre, compatibles con *Eucalyptus*, para la creación de máquinas virtuales. Una vez que se tiene una máquina virtual en ejecución, el usuario puede utilizarla según su conveniencia. Para acceder a los recursos que ofrece el prototipo se utilizan herramientas de línea de comandos denominadas *euca2ools* o una interfaz web, es decir, el manejo de dichos recursos se lo hace a través de un cliente SSH o la herramienta de administración de *Cloud Computing* denominada *Elasticfox*. Adicionalmente, el usuario puede crear volúmenes lógicos de almacenamiento, para tener un respaldo de su información y servicios implementados en las máquinas virtuales, estos recursos almacenados pueden ser utilizados en otra máquina virtual simplemente al asociarse con el volumen lógico que los contiene. También se pueden crear respaldos de dichos volúmenes, éstos se denominan *snapshots*.

Finalmente, hay que mencionar que las pruebas de funcionamiento del prototipo se han realizado a través de máquinas físicas y máquinas virtuales que ejecutan sistemas operativos como Windows y diferentes distribuciones de Linux respectivamente, a través de las cuales varios usuarios pueden acceder a la administración de los recursos que ofrece la Nube utilizando sus correspondientes credenciales.

PRESENTACIÓN

En el presente proyecto se muestran algunas definiciones, características y términos relacionados con *Cloud Computing*, sus modelos de infraestructura y de servicios, junto con la información correspondiente a la implementación de dicho prototipo y las pruebas de funcionamiento por parte del usuario. Entre los objetivos específicos planteados para este proyecto se pueden mencionar:

Implementar el prototipo de *Cloud Computing* de modelo privado, utilizando el *middleware Eucalyptus* que permite ofrecer Infraestructura como Servicio (IaaS).

Acoplar *Elasticfox* como plataforma de gestión de *Cloud Computing*, para administrar el prototipo; esta herramienta de gestión permite que los usuarios manejen todos los recursos de la Nube dependiendo de los permisos otorgados a cada uno de ellos.

Utilizar imágenes de diferentes sistemas operativos basados en software libre, y compatibles con *Eucalyptus*, para crear máquinas virtuales que se ejecuten dentro del prototipo.

Instalar un servidor web que se ejecute en una máquina virtual dentro del prototipo de *Cloud Computing* de modelo privado, para que mediante la realización de las pruebas correspondientes, se pueda comprobar su correcto funcionamiento.

Permitir que múltiples usuarios utilicen un navegador web o cliente SSH, para acceder a los servicios ofrecidos por el prototipo de *Cloud Computing* de modelo privado.

Tanto el marco teórico, como los detalles de la implementación y pruebas del prototipo planteado, se han incluido en tres capítulos modulares, la descripción del contenido de cada uno de éstos se la presenta a continuación:

En el Capítulo I se menciona una introducción teórica general del concepto de *Cloud Computing*, se describe su definición y taxonomía, su arquitectura y modelos de infraestructura, modelos de servicios, ventajas y desventajas, y su relación con otras tecnologías.

En el Capítulo II se da una breve introducción de las herramientas de software que se utilizan, se especifican los requerimientos de hardware, diseño de la topología de red y ubicación de los componentes físicos, se describe también la implementación del prototipo de *Cloud Computing* de modelo privado. Adicionalmente, se presenta el proceso de inserción de imágenes de diferentes sistemas operativos compatibles con el *middleware Eucalyptus*, se configura *Elasticfox* como herramienta de gestión de *Cloud Computing* a la cual el usuario tiene acceso desde el navegador web *Mozilla Firefox* y, finalmente, se indica el costo de los elementos utilizados para la realización del proyecto.

En el Capítulo III se describe el manejo de diferentes cuentas de usuario utilizando un servidor de correo electrónico, y el proceso de creación de volúmenes lógicos valiéndose del servicio denominado *Block Storage Service* que permite el almacenamiento persistente de datos al usar el *middleware Eucalyptus*. Se instala e implementa un motor de base de datos *mysql* y un servidor web, que se ejecutan en una máquina virtual instanciada por el usuario, para comprobar el funcionamiento de los servicios ofrecidos por el prototipo de *Cloud Computing* de modelo privado.

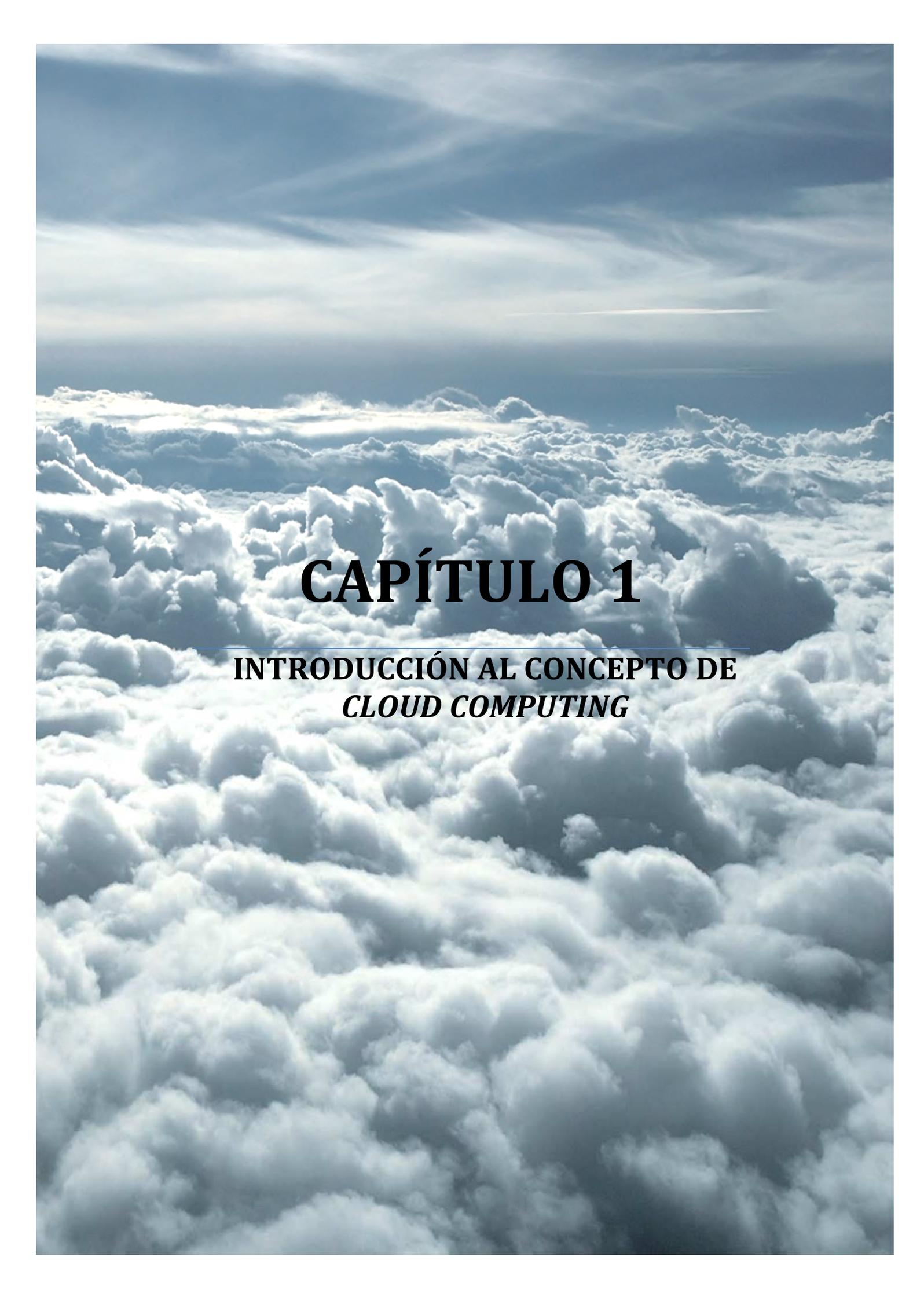
En el Capítulo IV se presentan conclusiones y recomendaciones basadas en experiencias que se obtuvieron durante el desarrollo del presente proyecto.

De forma adicional, los Anexos A y B presentan información sobre los proveedores de servicios de *Cloud Computing*, características de virtualización y una lista de *hypervisors* que se podrían utilizar para la implementación de *Cloud Computing*.

Los Anexos C y D, proporcionan características de la especificación Dual Core e información adicional acerca de *Amazon Web Services* (AWS) y la relación de éstos con *Eucalyptus*.

Los Anexos E y F ofrecen configuraciones de la BIOS en los nodos e información sobre los costos de los equipos utilizados en la implementación del prototipo.

El Anexo G, presenta información acerca de una herramienta de gestión de la Nube alternativa a la herramienta de gestión *Elasticfox* presentada en el proyecto. El Anexo H brinda información de las configuraciones necesarias para el monitoreo del tráfico generado desde y hacia la Nube utilizando MRTG (*Multi Router Traffic Grapher*).



CAPÍTULO 1

**INTRODUCCIÓN AL CONCEPTO DE
*CLOUD COMPUTING***

1.1 INTRODUCCIÓN

En los últimos años, el desarrollo de las Tecnologías de Información (TI) ha dado paso a que los datos, la música, el video, etc. se encuentren almacenados y distribuidos a través del Internet, de tal manera que dichos recursos puedan ser accesibles en cualquier momento y, prácticamente, desde cualquier lugar.

Los sistemas operativos han sido siempre el corazón de los servidores y de los computadores personales, y a lo largo del desarrollo de la tecnología, los nuevos elementos de software han ido adaptándose a los componentes de hardware para contar con un rendimiento cada vez más alto en el uso de aplicaciones. Sin embargo, los computadores no son utilizados todo el tiempo, a menos que se trate de brindar un servicio específico que demande una alta disponibilidad (24x7), e incluso en estos casos realizan tareas que no requieren de toda su capacidad disponible.

Por otro lado, existen servidores antiguos, con sistemas operativos antiguos, y que ejecutan aplicaciones antiguas. Para migrar a las nuevas aplicaciones que exige el desarrollo de las TI es necesario seguir un largo proceso. *“La mayoría de departamentos de TI se ven obligados a dedicar una buena parte de su tiempo a la tarea de implementar, mantener y actualizar proyectos, que con demasiada frecuencia, no suponen un valor añadido en el balance final de la empresa”* [1] y que en un determinado momento se verán obsoletos.

Diversos estudios muestran que aproximadamente el 75% del presupuesto de TI de una compañía se gasta en mantenimiento de la estructura existente, mientras que tan sólo el 25% se dedica a innovación. [2]

Es así que debido a la necesidad de acelerar este proceso y además de aprovechar al máximo los recursos en los que se ha invertido, nace lo que se conoce como la virtualización.

La virtualización es un método que se utiliza para ejecutar sistemas operativos múltiples e independientes en una sola máquina física. [3]

La tecnología de virtualización aprovecha los recursos de hardware subutilizados y la Computación en Nube (*Cloud Computing*) la toma como base para poder ofrecer servicios de infraestructura, plataforma y software, a clientes que solamente requieren de estos servicios bajo demanda, permitiendo al usuario pagar al proveedor de este servicio, únicamente por lo que consume.

Es así como *Cloud Computing* se incorpora a las TI, minimizando el tiempo empleado en actividades de implementación y centrándose en actividades más estratégicas, que tienen un mayor impacto en los procesos comerciales.

La Tabla 1.1 muestra la diferencia de tiempo empleado para realizar la misma tarea de implementación utilizando la tecnología tradicional y *Cloud Computing* por separado. Se observa que con *Cloud Computing* se consiguen menores tiempos.

Tareas de gestión de TI	Implementación Tradicional	Implementación <i>Cloud Computing</i>
Asignación de Servidores	3 días	< 1 hora
Instalación de software	5–10 días	< 1 hora
Configuración de red y parámetros de seguridad	5–10 días	< 1 hora
Sacar respaldos del sistema operativo	2 horas	1/2 hora
Recuperación del sistema operativo	2 horas	1 hora
Instalación de parches al sistema operativo	2 horas	1 hora
Asignación dinámica de recursos	1 horas	2 minutos
Regulación de parámetros en el sistema operativo, para diferentes servicios	10 minutos	1 minuto
Total	14–24 días	< 6 horas

Fuente: [17]

Tabla 1.1 Implementación tradicional vs implementación en *Cloud Computing*

Esta comparación fue realizada por IBM en la implementación de servidores utilizando el modelo tradicional y el modelo basado en *Cloud Computing*, tomando en cuenta ocho tareas de gestión de TI.

La flexibilidad del acceso a los recursos permite utilizar simplemente un sistema operativo “JeOS”¹ que ejecute un navegador web para poder utilizar o desarrollar aplicaciones. Ya no existiría la necesidad de instalar software de escritorio en los sistemas operativos.

Más adelante se explicará la relación que existe entre la virtualización y *Cloud Computing* ya que dentro de una empresa, son tecnologías que pueden adoptarse juntas o de forma individual. La virtualización es parte fundamental de *Cloud Computing* pues gracias a ella es posible disponer de todos los servicios que *Cloud Computing* ofrece.

1.2 DEFINICIÓN DE *CLOUD COMPUTING*

Las siguientes son definiciones de *Cloud Computing* en un entorno de una red privada y una pública.

- a) “*Capacidades de Tecnologías de Información (TI) escalables que se entregan como un servicio a usuarios externos usando tecnologías de Internet*”.²
- b) “*Cloud Computing, es la entrega de recursos de cómputo a través de una interfaz web (SOAP³ o REST⁴) según las necesidades.*

¹ **JeOS (Just Enough Operating System)** Sistema operativo básico, ofrece únicamente los componentes necesarios para hacer que una aplicación se ejecute; sin interfaces innecesarias, funciones, librerías ni servicios.

² Definición según Gartner. [4]

³ **SOAP (Simple Object Access Protocol)** Protocolo de acceso a objetos simple, es una especificación de protocolo de intercambio de información estructurada en la aplicación de Servicios Web en redes informáticas.

⁴ **REST (Representational State Transfer)** Transferencia de representación del Estado, es una técnica de arquitectura de software para sistemas hipermedia distribuidos como la *World Wide Web*.

El término Nube se refiere a la organización de infraestructura física, la cual es transparente para el usuario final.

Cloud Computing da al usuario acceso a los recursos de cómputo (máquinas, almacenamiento, sistemas operativos, entornos de desarrollo de aplicaciones, programas) a través de una red (Internet o Intranet) valiéndose de los Servicios Web. La organización física de los equipos que ofrecen estos recursos, puede hallarse ya sea en la habitación contigua o distribuidos alrededor del mundo, ya que no es necesario que sea de conocimiento del usuario pues esos recursos aparecen ante el usuario como si se encontraran alojados en la (Nube)"⁵. [7]

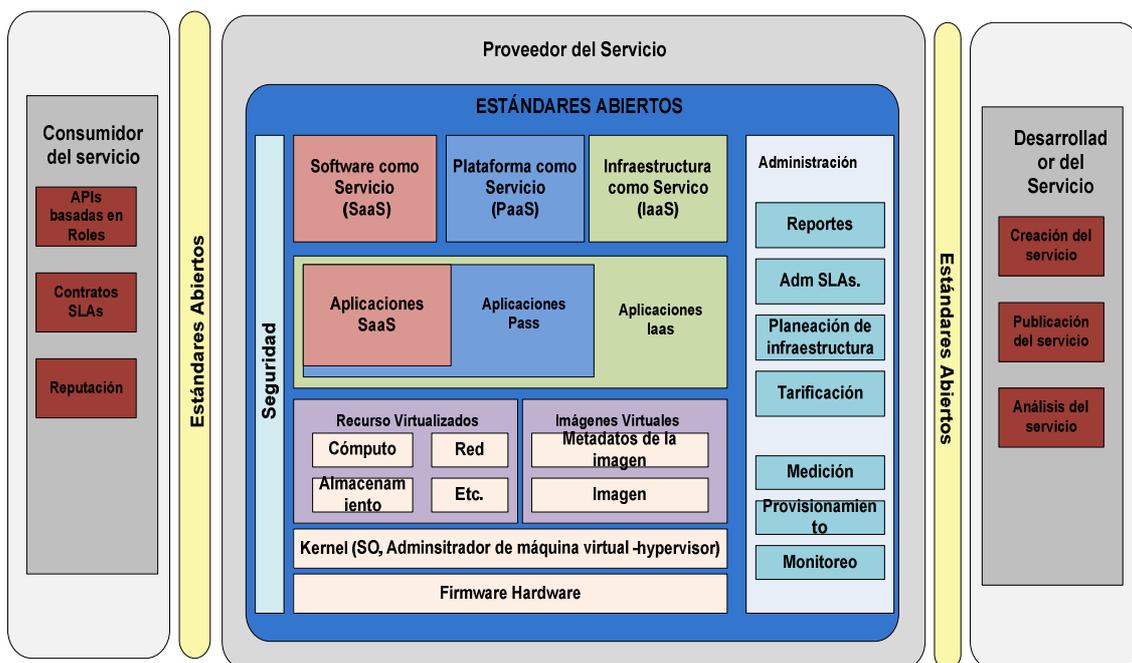
Cloud Computing es un modelo de computación, que permite ofrecer, de manera escalable, cualquier tipo de recurso virtualizado como un servicio a través de una red.

Cloud Computing puede ofrecer servicios utilizando Centros de Datos (*data centers*) propios dentro de una empresa, o se pueden utilizar los servicios ofrecidos por la Nube⁶ de un proveedor que es quien tiene el control de los elementos físicos de su infraestructura.

En la Figura 1.1 se pueden observar varios conceptos que son abarcados por *Cloud Computing*. Algunos de estos serán tomados en cuenta para explicar cómo funcionan las Nubes.

⁵ Definición de *Eucalyptus Systems*, en el segundo capítulo se tratará acerca de esta compañía.

⁶ **Nube (Cloud)**, en este proyecto se utilizará también este término para hacer referencia a la Computación en Nube (*Cloud Computing*).



Fuente: [36]

Figura 1.2 Taxonomía para *Cloud Computing*

1.3.1 CONSUMIDOR DEL SERVICIO (*SERVICE CONSUMER*)

El consumidor de los servicios, ubicado a la izquierda de la Figura 1.2, es el usuario final o empresa que en realidad utiliza el servicio, ya sea software, plataforma o infraestructura como servicio.

Dependiendo del tipo de servicio y su función, el consumidor trabaja con diferentes interfaces de usuario e interfaces de programación. El consumidor no necesita saber acerca de *Cloud Computing* ya que las interfaces son parecidas a las de escritorio. Otras interfaces de usuario proporcionan funciones administrativas, tales como iniciar y detener las máquinas virtuales o gestionar el almacenamiento de las Nubes.

Los consumidores escriben el código de una aplicación usando diferentes interfaces dependiendo de la aplicación que estén escribiendo. Trabajan con

los acuerdos de nivel de servicio (*Service Level Agreement, SLA*⁸) y los contratos. Generalmente, éstos son negociados entre el consumidor y el proveedor. Las expectativas del consumidor y la reputación del proveedor son una parte clave de esas negociaciones.

El desarrollador del servicio crea, edita y supervisa los servicios que ofrecen las Nubes; por lo general, las aplicaciones que se entregan directamente a los usuarios finales a través del modelo SaaS (*Software as a Service*) son "línea de negocio". Las aplicaciones escritas en IaaS (*Infrastructure as a Service*), y los niveles de PaaS (*Platform as a Service*) serán posteriormente utilizados por los desarrolladores y proveedores de SaaS en la Nube. [36]

1.3.2 PROVEEDOR DEL SERVICIO (*SERVICE PROVIDER*)

El proveedor del servicio es quien lo entrega al consumidor. Los detalles de estas tareas se especifican más adelante, ya que, como se puede apreciar en la Figura 1.2, existen diferentes tipos de servicios y otros aspectos que deben tomarse en cuenta para poder ofrecerlos.

En Software como Servicio (Figura 1.2, *Software as a Service, SaaS*), el proveedor instala, gestiona y mantiene el software. El proveedor no necesariamente es dueño de la infraestructura donde se ejecuta el software, pero independientemente de eso, el consumidor no tiene acceso a esa infraestructura, sólo a las aplicaciones.

Para Plataforma como Servicio (Figura 1.2, *Platform as a Service, PaaS*), el proveedor gestiona la infraestructura para la plataforma de las Nubes, por lo general un *framework*⁹, para un tipo concreto de aplicación. El consumidor no puede acceder a la infraestructura por debajo de la plataforma.

⁸ **SLA (*Service Level Agreement*)** es un documento habitualmente anexo al Contrato de Prestación de Servicios. En el SLA se estipulan las condiciones y parámetros que comprometen al prestador del servicio (habitualmente el proveedor) a cumplir con unos niveles de calidad de servicio frente al contratante de los mismos (habitualmente el *cliente*).

⁹ **Framework** (marco de aplicación) es un conjunto de bibliotecas orientadas a la reutilización a muy gran escala de componentes software para el desarrollo rápido de aplicaciones.

Para Infraestructura como Servicio (Figura 1.2, *Infrastructure as a Service, IaaS*), el proveedor mantiene el almacenamiento, la base de datos, *middleware*¹⁰ o el entorno de alojamiento de las máquinas virtuales. El consumidor utiliza dicho servicio como si fuera una unidad de disco duro, bases de datos, o una máquina, pero no puede acceder a la infraestructura que aloja este servicio.

Estos tres tipos de servicios, serán retomados adelante para tratarlos con mayor detalle en la Sección 1.4.

En la Figura 1.2, en el diagrama del proveedor de servicios, la capa inferior de la pila es el *firmware* y hardware en el que se basa todo lo demás. Por encima del *firmware* está el núcleo de software (*kernel*), ya sea el sistema operativo o el gestor de la máquina virtual (*Virtual Machine Manager, VMM*) que aloja la infraestructura por debajo de la Nube. Los recursos virtuales y las imágenes incluyen los servicios en la Nube de computación básicos tales como la potencia de procesamiento, almacenamiento y *middleware*. Las imágenes virtuales controladas por el VMM incluyen tanto las propias imágenes, así como los metadatos necesarios para su gestión.

La capa de gestión es crucial para las operaciones del proveedor de servicios. A nivel bajo, la gestión requiere de: medición para determinar quién utiliza los servicios y en qué medida, el aprovisionamiento para determinar cómo se asignan los recursos a los consumidores, y monitoreo para el seguimiento del estado del sistema y sus recursos.

A alto nivel, la gestión incluye la facturación para recuperar los costos, la gestión de SLAs para asegurar que los términos del servicio convenido entre el proveedor y el consumidor se cumplan, y la presentación de informes a los administradores para llevar mayor control acerca del proceso.

¹⁰ **Middleware** es un *software* de conectividad entre la capa de aplicaciones y las capas inferiores (sistema operativo y red), que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas.

La seguridad y los estándares se aplican a todos los aspectos de las operaciones del proveedor de servicios. Un conjunto bien definido de normas y estándares abiertos simplifican las operaciones de la Nube del proveedor y la interoperabilidad con Nubes de otros proveedores. [36]

1.3.3 DESARROLLADOR DEL SERVICIO (*SERVICE DEVELOPER*)

El desarrollador de servicio, ubicado en la parte derecha de la Figura 1.2, crea, edita y supervisa el servicio prestado por las Nubes.

Los entornos de desarrollo para la creación de servicios varían. Si los desarrolladores están creando una aplicación SaaS, lo más probable es que estén escribiendo código para un entorno hospedado por un proveedor de *Cloud Computing*. En este caso, la publicación del servicio está desplegándose desde la infraestructura del proveedor de la Nube.

Durante la creación de servicios, el análisis de éstos, implica la depuración remota para probar el servicio antes de su publicación para los consumidores. Una vez que el servicio es publicado, el análisis permite a los desarrolladores supervisar el rendimiento de su servicio y posteriormente hacer cambios si fuera necesario. [36]

1.4 MODELOS DE INFRAESTRUCTURA EN *CLOUD COMPUTING*

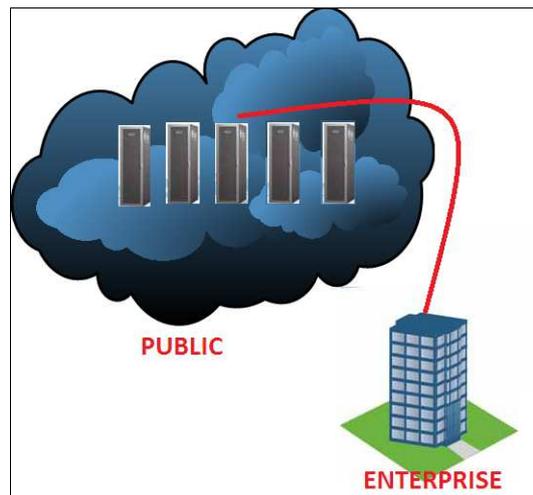
Se deben considerar varios aspectos a la hora de tomar la decisión de implementar un modelo de *Cloud Computing*. Existen tres modelos básicos a considerar, los cuales pueden ser de origen propietario, o basados en software libre.

1.4.1 *CLOUD COMPUTING* DE MODELO PÚBLICO

El proveedor de los servicios de *Cloud Computing* es dueño de la infraestructura física y pone a disposición del cliente los servicios de la Nube a

través del Internet; ésta es su característica principal, y es la que permite que el usuario pueda acceder a dichos servicios en cualquier momento y lugar.

En la Figura 1.3, la infraestructura que utiliza el proveedor para ofrecer los servicios, puede estar alojada en cualquier lugar del Internet, y el consumidor accede a dichos servicios utilizando una conexión a Internet.



Fuente: [13]

Figura 1.3 *Cloud Computing* de modelo público

Ya que los servicios de la Nube se encuentran disponibles a través del Internet, el cliente no puede saber exactamente en donde se está ejecutando su aplicación ni cuantos equipos forman parte de la infraestructura.

El usuario tendrá el control de determinados recursos y puede conectarse a través de un navegador web o un cliente SSH, dependiendo del tipo de servicio que utilice.

De acuerdo al uso de los servicios ofrecidos por la Nube, el usuario tendrá que pagar únicamente por lo que consuma. Esto representa un gran ahorro en la inversión inicial de un proyecto, puesto que no se tiene que comprar toda la infraestructura para desplegarlo.

Quien contrata el servicio no se preocupa del mantenimiento, actualización, licencias, etc., todo esto está a cargo del proveedor de los servicios de la Nube,

y en base a los Acuerdos de Nivel de Servicio, SLA, establece los niveles de servicio al cliente.

Una Nube pública provee servicios a múltiples clientes, y es típicamente implementada en un centro controlado, instalaciones seguras, con vigilancia de 24 horas y un sistema de alimentación ininterrumpida.

A medida que las necesidades del usuario se incrementan, eventualmente, éste podría requerir mayores prestaciones del proveedor, lo cual no representa ningún riesgo en el rendimiento ya que el modelo permite contar con escalabilidad de una manera sencilla.

Los sistemas en la Nube controlan y optimizan el uso de los recursos de manera automática, por tanto, el uso de éstos puede seguirse, controlarse y notificarse, lo que aporta transparencia tanto para el proveedor como para el consumidor del servicio utilizado. [11]

Al ser un sistema establecido en una red pública, la seguridad es un aspecto muy importante; actualmente, se utilizan métodos para transmitir la información cifrada pero hay que tomar en cuenta que los datos no se encuentran almacenados localmente. Esta característica es lo que ha promovido que se cree otro modelo de Nube: *Cloud Computing* privada. A pesar de ello, los proveedores cuentan con *data centers* específicamente dedicados a la custodia y salvaguarda de datos ya que han implementado todas las medidas de seguridad necesarias, tanto de *hardware* como de software, de forma que no haya jamás una pérdida de información ni de integridad de los datos. La única precaución que hay que tener, es encontrar el proveedor que ofrezca las garantías y prestaciones adecuadas a los servicios que se requieren.

Dependiendo de la organización en la que se desee implementar *Cloud Computing*, pueden existir casos en los que se requiera grandes capacidades de cómputo y en los que la Nube pública resulte costosa.

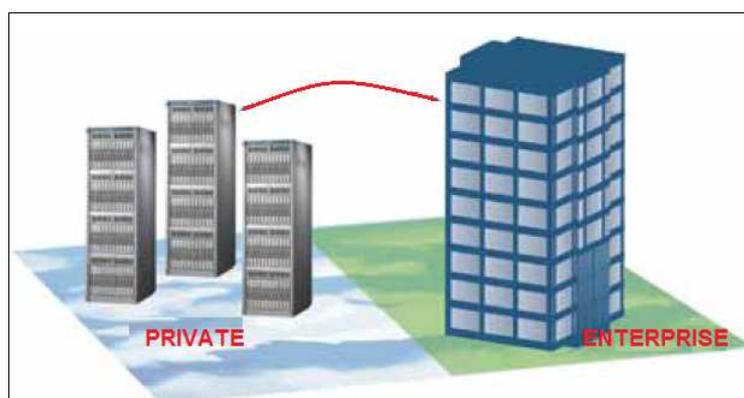
1.4.2 CLOUD COMPUTING DE MODELO PRIVADO

Es una emulación de la Nube pública, pero en una red privada. Ofrece los mismos servicios que una Nube pública con la ventaja de que el usuario cuenta con sus propios recursos, lo que le permite tener el control total de seguridad y Calidad de Servicio (*Quality of Service, QoS*), sobre ellos.

Entre los requisitos básicos de una Nube privada se pueden mencionar: medición y control, gestión, seguridad, despliegue o implementación, interoperabilidad y un formato de máquina virtual común.

Una Nube privada no requiere estándares de la industria, ni APIs¹¹ comunes entre otras Nubes para el *middleware* que se ha utilizado en su implementación.

Las Nubes privadas pueden ser implementadas directamente por el propietario o por un proveedor. En este caso el proveedor, únicamente, se encarga de instalar todos los elementos necesarios para que el cliente disponga de la infraestructura de *Cloud Computing*, en su propio *data center*. La otra opción consiste en que el usuario implemente su propia Nube, para esto puede usar sistemas basados en software libre.



Fuente: [13]

Figura 1.4 *Cloud Computing* de modelo privado

¹¹ API (*Application Programming Interface*) es la interfaz de programación de aplicaciones, representa una interfaz de comunicación entre componentes de software.

Las Nubes privadas pueden ser implementadas en un *data center* corporativo. Pueden obtener el soporte de la misma compañía, de un proveedor de *Cloud Computing*, o de un tercero como una firma de *outsourcing*¹².

Las Nubes privadas permiten incrementar el nivel de seguridad de los datos considerados como sensibles, pues la infraestructura está bajo el control de la misma organización que la va a utilizar.

En la Figura 1.4, la infraestructura utilizada para ofrecer servicios se encuentra dentro de los dominios de la empresa, lo que se percibe como un entorno más seguro para el dueño de los datos.

La escalabilidad en este modelo sigue presentándose de una manera sencilla pues en el caso de que el sistema implementado esté utilizando al máximo sus recursos, se puede contar con la interconexión a una Nube pública, lo cual se conoce como *Nube híbrida* o *Cloud Computing* de modelo híbrido. De esta manera se pueden solucionar picos de demanda¹³, sin la necesidad de invertir en más recursos de *hardware* que sólo serán utilizados en determinados momentos.

En este proyecto se utilizará el *middleware* conocido como *Eucalyptus*, al cual se hará referencia en capítulos posteriores, para implementar una Nube privada.

1.4.3 CLOUD COMPUTING DE MODELO HÍBRIDO

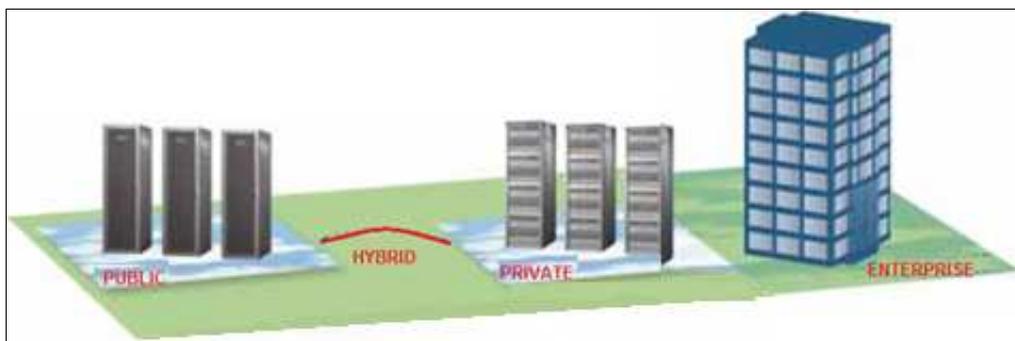
La Nube híbrida combina los modelos público y privado. Tiene la ventaja de contar con los beneficios de ambos modelos, lo cual permite aumentar la capacidad de una Nube privada con los recursos de una Nube pública para

¹² **Outsourcing** o tercerización (también llamada subcontratación) es una técnica de administración, que consiste en la transferencia a terceros de ciertos procesos complementarios que no forman parte del giro principal del negocio, permitiendo la concentración de los esfuerzos en las actividades esenciales a fin de obtener competitividad y resultados tangibles.

¹³ **Pico de demanda** define un aumento temporal o aleatorio de la demanda de algún producto o servicio.

poder mantener niveles de servicio adecuados frente a rápidas fluctuaciones de carga de trabajo.

La Figura 1.5 muestra la conexión de una empresa con una Nube híbrida, teniendo también dentro de sus dominios una Nube privada.



Fuente: [13]

Figura 1.5 *Cloud Computing* de modelo híbrido

Las Nubes híbridas pueden ser particularmente efectivas cuando ambos modelos de Nube, están ubicados en la misma localidad.

Las Nubes híbridas introducen la complejidad de determinar cómo distribuir aplicaciones entre las Nubes pública y privada. Entre las cuestiones que se deben tener en cuenta está la relación entre los datos y recursos de procesamiento.

Si la cantidad de datos que se maneja es pequeña, una Nube híbrida puede ser mucho más conveniente, ya que esos datos pueden ser almacenados y procesados ya sea en la Nube privada o utilizando la Nube pública. En el caso de manejar grandes volúmenes de datos y que se requiera una pequeña capacidad de procesamiento, es aconsejable utilizar una Nube privada, pues no se justificaría contratar un gran ancho de banda para transferir los datos a la Nube pública para que ésta los procese. [13]

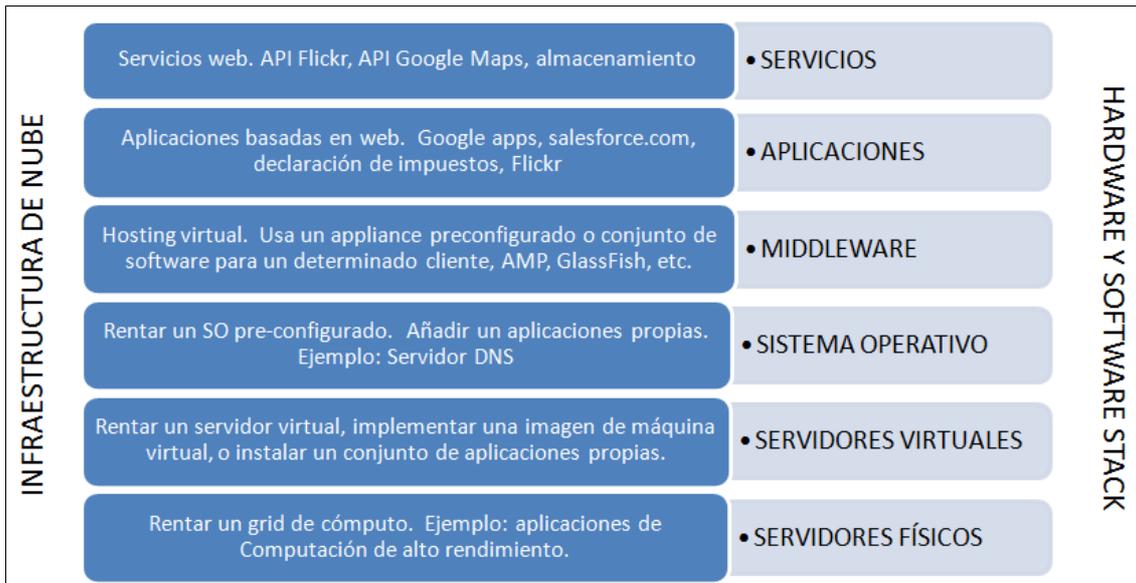
1.5 ARQUITECTURA EN CAPAS Y MODELOS DE SERVICIO DE *CLOUD COMPUTING*

Cloud Computing tiene la capacidad de ofrecer cualquier tipo de servicio basándose en la estructura tradicional de capas que va desde el hardware hasta las aplicaciones. Los servicios que entregan las Nubes son útiles para cualquier tipo de consumidor o entidades de negocios.

Los ofrecimientos de *Cloud Computing* son relativamente amplios, ya que los clientes se conectan a los diversos servicios a través de una red (utilizando un navegador web o cliente SSH), sin importar el tipo de dispositivos utilizados y la localización de los usuarios, lo cual implica un ahorro de recursos al utilizar dispositivos antiguos que tal vez ya no tenían buen uso pues, para conectarse con la Nube no se necesita de un sistema operativo completo.

El Internet está sirviendo como una muy buena solución para situaciones en las que es más importante obtener un resultado, que el propio hardware o software que se necesita para conseguirlo. El uso de navegadores web ha dado lugar a una migración constante del modelo tradicional de *data center* a un modelo basado en la Nube.

En la Figura 1.6 se pueden observar las capas en las que se puede obtener el servicio de *Cloud Computing*. Desde el alquiler de servidores físicos (hardware) hasta el uso de aplicaciones (software).



Fuente: [13]

Figura 1.6 Componentes y capas de Cloud Computing

Cloud Computing implica el uso de la infraestructura de TI como un servicio, y dicho servicio puede ser de cualquier tipo, desde hardware, hasta APIs de terceros.

Desde el punto de vista del proveedor (Figura 1.7), la arquitectura de *Cloud Computing* comprende tres capas. A continuación se explica detalladamente cada una de ellas y los diferentes servicios ofrecidos por la capa superior.



Fuente: [Elaboración propia]

Figura 1.7 Arquitectura en capas y Modelo de servicios en *Cloud Computing*

1.5.1 HARDWARE

De acuerdo a la Figura 1.7, esta capa está en la parte inferior de la arquitectura de capas de *Cloud Computing*, contiene a todos los dispositivos físicos que hacen funcionar a la Nube y envuelve a un sin número de ordenadores apilados unos sobre otros, por lo general dentro de *data centers*. [80]

Algunos de los ordenadores estarán destinados a almacenar datos y archivos, y otros a procesar información. Todos ellos son fácilmente reemplazables y si se requiere ampliar la capacidad de la Nube, tan sólo se añaden más ordenadores o se puede construir un nuevo *data center*, si los recursos económicos son suficientes.

1.5.2 VIRTUALIZACIÓN

En la Figura 1.7, la capa de virtualización se encuentra entre la capa de hardware y la capa de servicios.

La virtualización es un término amplio que se refiere a la abstracción de los recursos de una computadora. Este término viene siendo utilizado desde antes de 1960, y ha sido aplicado en diferentes aspectos de la informática, desde sistemas operativos completos hasta componentes individuales.

La virtualización de la plataforma crea una interfaz externa que esconde una implementación subyacente mediante la combinación de recursos en ubicaciones físicas diferentes, o mediante la simplificación del sistema de control [2].

La virtualización involucra la simulación de una máquina real, la cual está alojada en un sistema anfitrión (*host*) y se ejecuta a través de un software llamado *hypervisor*¹⁴, el cual controla que todas las máquinas virtuales sean atendidas correctamente. El *hypervisor* crea una capa de abstracción entre el

¹⁴ **Hypervisor o VMM (Virtual Machine Monitor)** es una plataforma de virtualización que permite utilizar, al mismo tiempo, diferentes sistemas operativos o máquinas virtuales (sin modificar o modificados) en una misma computadora central.

hardware de la máquina física (*host*) y el sistema operativo de la máquina virtual (*virtual machine guest*) de tal forma que maneja los recursos de las máquinas físicas subyacentes para que el usuario pueda crear varias máquinas virtuales presentando a cada una de ellas, una interfaz del *hardware* que sea compatible con el sistema operativo elegido. [14]

Existen otros tipos de virtualización, los cuales se pueden consultar en el Anexo B, para mayor información.

1.5.2.1 Virtualización y *Cloud Computing*

Cuando un usuario solicita una colección de máquinas virtuales (MV) a una Nube, las MVs deben estar completamente aisladas de las otras en términos de acceso al CPU, memoria, red y almacenamiento persistente.

Todas estas formas de acceso aislado deben estar asociadas a un usuario autenticado con propósitos de seguridad. La virtualización del sistema operativo y los *hypervisors* proporcionan un aislamiento no autenticado de CPU y memoria, pero no de la red privada entre máquinas virtuales ni almacenamiento persistente por cada usuario. [18]

Cloud Computing permite tener el control del acceso de cada usuario tanto a la red como a la máquina virtual, y dependiendo del servicio que utilice en una Nube tendrá el control de determinados recursos (revisar Sección 1.3).

“En la actualidad, la Virtualización y Cloud Computing están siendo aceptados y adaptados por un número creciente de usuarios. Los proveedores de servicios y contenidos están constantemente desplegando nuevos servicios de Cloud Computing como el S3, EC2 de Amazon, o Synaptic Hosting de AT&T, proporcionando entornos de TI completamente gestionados, para que las empresas puedan cargar y ejecutar sus aplicaciones”. [19]

1.5.3 SERVICIOS

En la Figura 1.7 la capa de servicios se encuentra en la parte superior de la arquitectura de capas de *Cloud Computing*.

La computación de hoy en día se ha desarrollado de tal forma que todo lo que se encuentra en un *data center* se pueda ofrecer como servicio. El término (*Anything-as-a-Service, XaaS*) o lo que sea como servicio da la pauta para encontrar un subconjunto de servicios que actualmente los proveedores de *Cloud Computing* ofrecen:

- Infraestructura como Servicio (*Infrastructure as a Service, IaaS*)
- Plataforma como Servicio (*Platform as a Service, PaaS*)
- Software como Servicio (*Software as a Service, SaaS*)

Los servicios ofrecidos por las Nubes pueden ser manejados de diferentes formas de acuerdo al modelo de infraestructura que se utilice, ya que como se mencionó anteriormente, en una Nube de modelo privado la empresa tiene el control de acceso, y control sobre toda la infraestructura que la conforma, lo que le permite disponer de servicios personalizados que se pueden adecuar totalmente a sus necesidades.

Por otro lado, el modelo público de *Cloud Computing*, es de propiedad y está controlado por un proveedor externo que proporciona acceso al mismo bajo suscripción. Por lo tanto, los servicios están definidos en un catálogo que no da margen a que sean personalizados. [17]

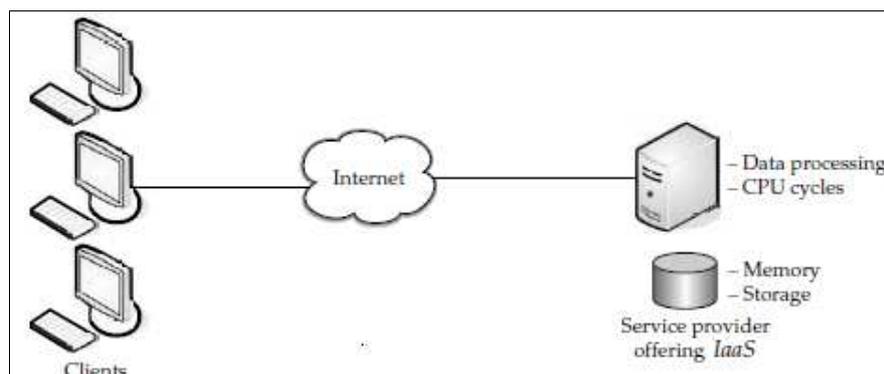
1.5.3.1 Infraestructura como Servicio (IaaS)

En Infraestructura como Servicio, la capacidad suministrada a los clientes es el abastecimiento de procesamiento, espacio de almacenamiento, equipos de red y otros recursos computacionales importantes para que los clientes puedan desplegar y ejecutar software de forma arbitraria, lo cual puede incluir sistemas operativos y aplicaciones. Se ofrecen diferentes equipos como servidores,

sistemas de almacenamiento, dispositivos de enrutamiento y otros que permiten manejar cargas de trabajo, que van desde los pequeños componentes hasta aplicaciones de computación de alto rendimiento.

La infraestructura se brinda, normalmente, mediante una plataforma de virtualización. Los clientes no gestionan ni controlan la infraestructura de la Nube, pero tienen control sobre los sistemas operativos, almacenamiento, aplicaciones desplegadas y la posibilidad de tener un control limitado de componentes de red seleccionados.

En la Figura 1.8 se muestran algunos recursos ofrecidos como servicio a los clientes que utilizan IaaS (procesamiento de datos, ciclos de CPU, memoria y almacenamiento).



Fuente: [16]

Figura 1.8 Infraestructura como Servicio

IaaS permite a los proveedores de servicios el alquiler de recursos de hardware.

Cliente:

- No invierte en infraestructura alguna.
- No incurre en gastos de operación.
- Paga sólo por el uso del servicio; es decir, el costo del servicio está basado en el tiempo y número de recursos que utilice.
- No existe ningún contrato de permanencia para el uso del servicio.

Arquitectura

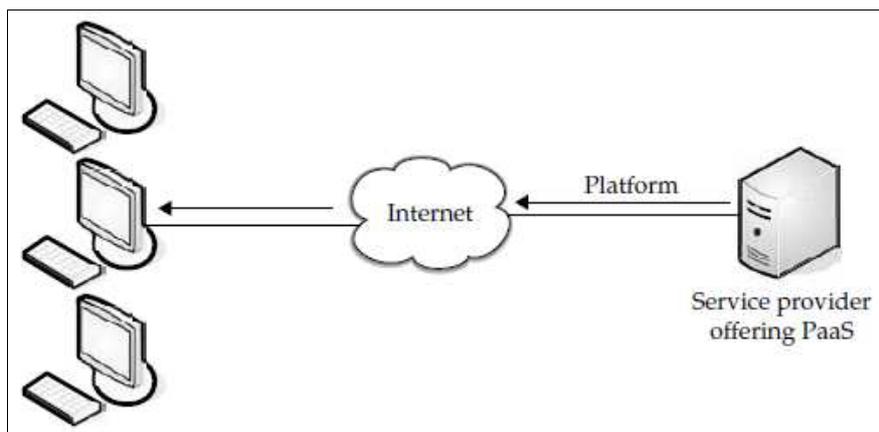
- Se maneja un escalado dinámico e inmediato, de acuerdo a la aplicación y a la necesidad de recursos a utilizarse.
- No importa la localización del usuario con respecto a la localización del proveedor del servicio.
- Compartir la capacidad entre múltiples usuarios para tener en cuenta los picos de carga y poder escalar rápidamente y sin dificultades, sumando así la reducción de las tareas de mantenimiento. [21]
- Usa tecnologías de virtualización, que permiten a los usuarios ejecutar el número de máquinas virtuales que ellos necesiten.

1.5.3.2 Plataforma como Servicio (PaaS)

En Plataforma como Servicio, la capacidad proporcionada a los clientes es desplegar en la infraestructura las aplicaciones adquiridas o creadas por ellos; es decir, el cliente puede desarrollar, probar e implementar sus aplicaciones en los centros de datos del proveedor, utilizando diferentes lenguajes y herramientas de programación que el proveedor del servicio soporte. En general, este tipo de servicio permite construir sistemas de alto nivel, ya que proporciona todos los recursos necesarios para crear aplicaciones y servicios desde Internet, sin tener que descargar o instalar el software. Los clientes no gestionan ni controlan la infraestructura de la Nube, pero tienen el control sobre las aplicaciones desplegadas y la posibilidad de controlar las configuraciones de entorno de alojamiento de las aplicaciones.

En la Figura 1.9 se esquematiza de forma general la manera en que los usuarios pueden acceder a PaaS.

PaaS permite a los clientes acceder a una plataforma de información sobre una solución de *Cloud Computing*.



Fuente: [16]

Figura 1.9 Plataforma como Servicio

Una desventaja de PaaS es la falta de interoperabilidad y portabilidad entre proveedores, es decir, si se crea una aplicación con un proveedor de Nube y se decide cambiar de proveedor, muchas veces el precio que se debe pagar es relativamente alto o al final es imposible hacerlo.

Además, si PaaS deja de funcionar, los datos y aplicaciones de los clientes también van a desaparecer para evitar este tipo de inconvenientes, es necesario controlar dos factores: la información de la empresa y el servicio. [22]

El servicio de plataforma ofrece cierto soporte de ayuda en la creación de interfaces de usuario basadas en HTML¹⁵ o *JavaScript*¹⁶. También soporta el desarrollo de interfaces web con SOAP y REST que permiten la construcción de múltiples Servicios Web, llamados *mashups*¹⁷.

Adicionalmente, este tipo de servicio ha sido diseñado para brindar facilidades de acceso a múltiples clientes simultáneos, manejo de concurrencia, escalabilidad, balanceo de carga y seguridad.

¹⁵ **HTML** o Lenguaje de Marcado de Hipertexto (*HyperText Markup Language*).

¹⁶ **JavaScript** es un lenguaje de programación interpretado, es decir que no necesita compilación ya que el lenguaje funciona del lado del cliente. Los navegadores web son los encargados de interpretarlos.

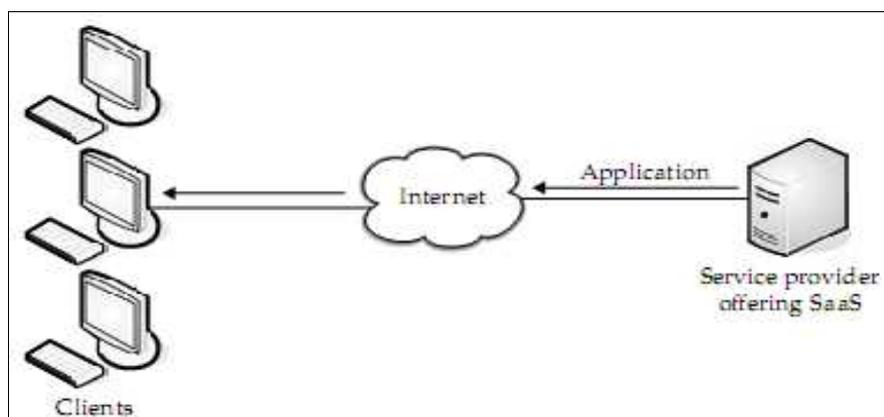
¹⁷ **mashup (aplicación web híbrida)** es una aplicación web que usa contenido de otras aplicaciones web para crear un nuevo contenido completo, consumiendo servicios directamente, y a través del protocolo http. El contenido de un *mashup* normalmente proviene de servicios web de terceros a través de una interfaz pública.

1.5.3.3 Software como Servicio (SaaS)

Es un modelo de distribución de software, el cual se encuentra alojado como una sola instancia, que se ejecuta en la infraestructura de la Nube y es distribuido a múltiples clientes a través de una red (pública o privada). El software se implanta dentro de la Nube y está disponible para el usuario a través de un navegador web, sin necesidad que el usuario lo adquiera, instale o realice algún tipo de mantenimiento, ya que el proveedor maneja todas las actualizaciones y el funcionamiento del servicio, por lo que queda fuera del alcance del cliente el sugerir al proveedor realizar cambios al servicio que presta.

SaaS provee una aplicación o una parte del software desde el proveedor de Servicios. La capacidad proporcionada a los clientes consiste en utilizar las aplicaciones del proveedor que se ejecutan en una infraestructura de Nube. Adicionalmente, el usuario no se preocupa dónde está instalado el software, qué tipo de sistema operativo utiliza o el lenguaje en el que éste está escrito.

En la Figura 1.10 se esquematiza, de forma general, la manera en que los usuarios pueden acceder a SaaS, de manera que el proveedor ofrece sus servicios a través de una red pública como el Internet.



Fuente: [16]

Figura 1.10 Software como Servicio

Este modelo ha permitido que se desarrollen tecnologías como los Servicios Web y SOA¹⁸; adicionalmente, existen varios tipos de software que no necesitan mucha interacción con otros sistemas, como la videoconferencia, manejo de Servicios de TI, manejo de contenido web.

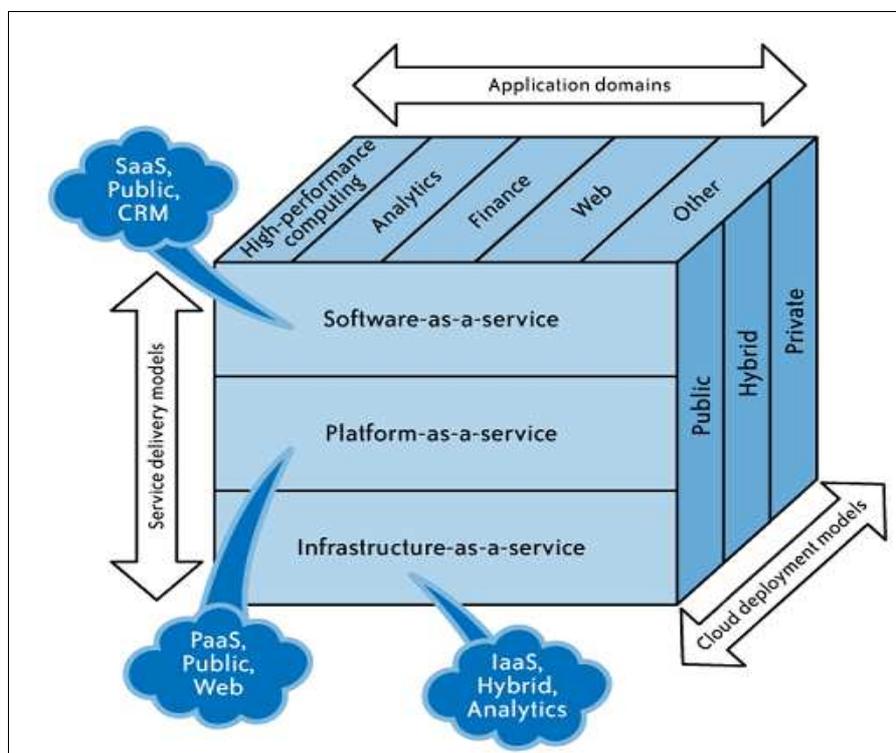
1.5.4 CARACTERÍSTICAS COMUNES DE LOS TRES SERVICIOS DE CLOUD COMPUTING

Los tres servicios que ofrece *Cloud Computing* manejan ciertas características comunes, como:

- Acceso al servicio a través de un navegador web, ya que los clientes no requieren la instalación de ningún software específico en sus computadores.
- Disponibilidad bajo demanda, ya que una vez que el usuario tenga el acceso, debería ser capaz de utilizar el software en cualquier momento y en cualquier lugar.
- Pago del servicio de acuerdo al uso del mismo, los usuarios no necesitan ninguna inversión en infraestructura o instalación, por lo que no se tiene que pagar por todos los costos de instalación. Simplemente, el cliente debe pagar por el tipo de servicio que utilice, durante el tiempo que lo emplee.
- Control de usuarios. El proveedor en *Cloud Computing* tiene una única aplicación que ofrece a todos los usuarios que requieren de su servicio, estableciendo acceso y prestaciones de recursos distintos para cada uno de ellos.

¹⁸**SOA (Service Oriented Architecture)** establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios.

- Monitoreo del servicio. Los sistemas basados en la Nube son monitoreados para controlar y optimizar el uso de los recursos.



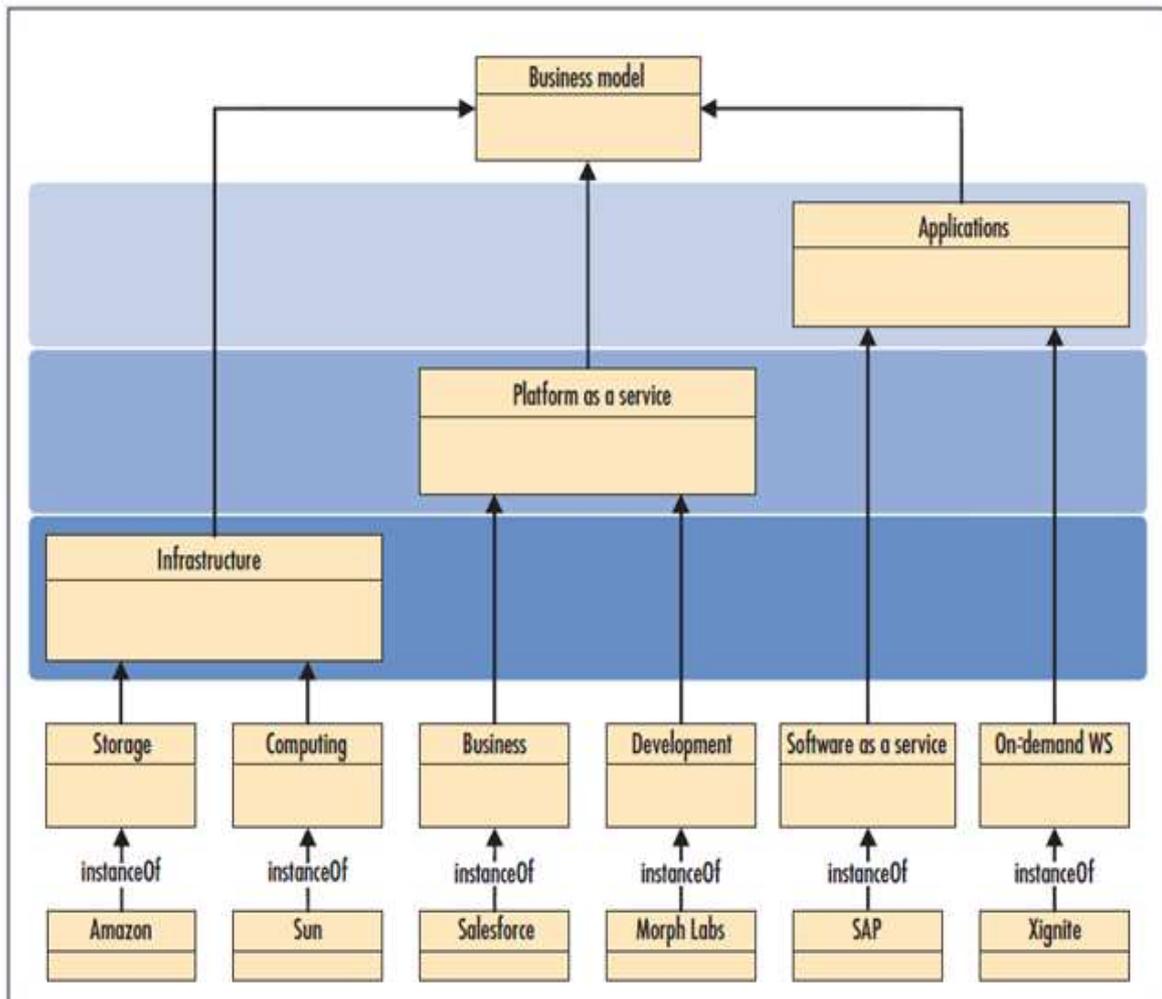
Fuente: [25]

Figura 1.11 Relación entre los servicios, aplicaciones y tipos de Nubes

Según la Figura 1.11, los tres niveles de servicio pueden ser ofrecidos utilizando *Cloud Computing* de modelo público, privado o híbrido, y su utilización puede ir desde aplicaciones web, hasta Computación de Alto Rendimiento (*High Performance Computing, HPC*)¹⁹.

En la Figura 1.12, la ubicación de los componentes es importante, los más altos facilitan la funcionalidad encapsulada de las capas inferiores mediante la agregación de los componentes de servicio a través de la composición y las tecnologías *mashup*. [27]

¹⁹ Tomar en consideración la Figura 1.2.



Fuente: [27]

Figura 1.12 Arquitectura de *Cloud Computing*

Los proveedores ponen a disposición del cliente, instancias de los recursos que serán ofrecidos como servicios (Figura 1.12) y que forman parte de un determinado modelo de negocios. Amazon (IaaS), Salesforce (PaaS), Xignite (SaaS).

En la Tabla 1.2 se detallan los distintos modelos de servicios que ofertan algunos de los proveedores, de *Cloud Computing* de modelo público, más utilizados.

Compañía /producto	Servicio	Forma de Pago	Modelo de servicio
Amazon EC2 y S3, SimpleDB, SQS, FPS, DevPay	Procesamiento, almacenamiento, bases de datos, facturación.	Pago por uso	Infraestructura / plataforma
Appian Anywhere	Gestión de procesos de negocio	Pago por uso	Aplicaciones o SaaS
Box.net	Almacenamiento	Pago por uso	Aplicaciones
FlexiScale	Infraestructura	Pago por uso	Infraestructura
Google App Engine	Infraestructura, aplicaciones web	Pago por uso	Infraestructura
Gmail Drive	Almacenamiento, correo electrónico	Pago por uso/ gratis	Aplicaciones
MuxCloud	Procesamiento de datos (video). Utiliza Amazon EC2	Pago por uso	Aplicaciones
Nirvanix	Almacenamiento	Pago por uso	Aplicaciones
Network.com	Infraestructura	Pago por uso	Infraestructura/ plataforma
OpSource	Facturación	Subscripción	Aplicaciones
Process Maker Live	Gestión de procesos de negocio	Pago por uso	Aplicaciones
Salesforce.com	Plataforma	Pago por uso	Plataforma / aplicaciones
MS SkyDrive	Almacenamiento	Gratis	Aplicaciones
SmugMug	Compartición de datos (fotos)	Subscripción	Aplicaciones
Strikelron	Servicios web	Subscripción /pago por uso	Aplicaciones
XDrive	Almacenamiento	Subscripción	Aplicaciones
XCalibre	Infraestructura	Subscripción	Infraestructura
Zimory.com	Mercado	Tarifación dinámica	Aplicaciones

Fuente: [28]

Tabla 1.2 Oferta de Servicios bajo demanda

Considerando un modelo privado, existen algunas soluciones de código abierto que permiten implementar más que nada, Infraestructura como Servicio, algunos de los más conocidos son: *Eucalyptus*²⁰ y *Open Nebula*²¹.

Existen opciones adicionales tanto de *middleware* para Nubes privadas como de proveedores de Nubes públicas, las cuales se pueden consultar en el Anexo A.

²⁰ Sitio web oficial: <http://www.eucalyptus.com/>.

²¹ Sitio web oficial: <http://www.opennebula.org/>.

1.6 VENTAJAS Y DESVENTAJAS DE *CLOUD COMPUTING*

Cuando una nueva tecnología se encuentra en las primeras fases de su despliegue, suele tener ventajas y desventajas que hay que tomar en cuenta a la hora de su implementación. En las Tablas 1.3 y 1.4 se muestran algunas de ellas, las cuales se han considerado relevantes para la realización de este proyecto. También se hará una distinción de cada una tomando en cuenta los modelos de Nube pública y privada.

En el caso de utilizar una Nube híbrida, las desventajas de una Nube pública ya no se presentan pues se combinan ambos modelos y se toma lo mejor de cada caso. Por ejemplo, en cuestiones de seguridad, la desventaja de usar una Nube pública era que los datos se almacenaban en un lugar remoto del cual el usuario no tenía control alguno; esto ya no sería un inconveniente ya que la parte privada de la Nube híbrida mantendría almacenados los datos localmente.

Características	<i>Cloud Computing</i> pública	<i>Cloud Computing</i> privada
Uso de computadores más baratos	Es necesario contar con un navegador web, pues todo el procesamiento, almacenamiento, uso de memoria RAM, se realiza en las instalaciones del proveedor de <i>Cloud Computing</i> . De esta manera, los usuarios no necesitan comprar un computador costoso que tenga excelentes características de hardware.	El proveedor de <i>Cloud Computing</i> está dentro de la misma organización que el usuario; por lo tanto, el usuario tendrá que conectarse a la Nube a través de un navegador web dentro de la red privada de la organización.
Menor costo de Infraestructura para TI	Cuando se produzcan picos de demanda, las organizaciones no necesitarán invertir en un gran número de servidores potentes para ayudar a combatirlos, y en su lugar podrán utilizar los servicios que <i>Cloud Computing</i> ofrece, para complementar o suplementar los recursos internos de cómputo.	Este modelo ofrece las mismas características. Si en un determinado momento los recursos de la Nube privada no son suficientes, se puede conectar con una Nube pública, la cual tomaría el nombre de Nube híbrida, para incrementar sus capacidades.
Menor costo de mantenimiento	<i>Cloud Computing</i> reduce significativamente los costos de mantenimiento, tanto de hardware como de software de organizaciones de todos los tamaños, ya que es el proveedor quien se ocupa de tales cuestiones.	En este caso, es la misma organización dueña de la infraestructura de la Nube; por tanto, los costos de mantenimiento corren por su cuenta. Es importante tomar en cuenta si es mejor adoptar <i>Cloud Computing</i> en un modelo público o en uno privado.
Menor tiempo en la instalación de software	El proveedor se encarga de la actualización e instalación de software en la Nube. De esta manera el usuario se ahorra la tarea de instalarlo y actualizarlo en cada máquina.	El usuario de la Nube no tiene la necesidad de instalar software para poder usar sus aplicaciones. El mantenimiento del <i>data center</i> donde está la infraestructura de la Nube, corre por cuenta del propietario en este caso la misma corporación.
Poder de Cómputo y Capacidad de almacenamiento Incrementados	Una Nube puede mantener un sinnúmero de recursos distribuidos y puede contar con una gran capacidad de almacenamiento que va escalando regularmente, lo que implica mayor capacidad disponible que un computador personal.	Dentro de este modelo, la capacidad y el poder de cómputo son los mismos instalados en el <i>data center</i> de la organización, así que el usuario a través de la red privada tendrá acceso a dichos recursos, los cuales en determinado momento, y con la ayuda de una conexión a una nube pública, podrían ser incrementados.
Seguridad en la recuperación de datos	Los datos se almacenan en un lugar incierto de la Nube. Los datos se respaldan automáticamente y si existen daños, la información se recupera inmediatamente y el daño corre por cuenta del proveedor.	Si el disco duro de un computador personal deja de funcionar, se perdería toda la información en él. En un <i>data center</i> de Nube privada se realizan respaldos, aunque la cuenta por el disco duro dañado corre por la misma organización.

Compatibilidad entre sistemas operativos	Para acceder a los servicios que una Nube ofrece no es necesario disponer de un sistema operativo en específico, ya que a pesar de que se utilice un navegador instalado en Windows, se pueden compartir archivos en la Nube con usuarios que estén usando un navegador web instalado en Linux o Mac OS.	De la misma forma, en la Nube privada el acceso es suficiente con un navegador web, sin importar la plataforma que se use.
Colaboración entre grupos más sencilla	Varios usuarios pueden tener acceso a la edición de un mismo documento simultáneamente, ya que dicho documento se encuentra almacenado en la Nube. Esto les permite a los usuarios encontrarse en ubicaciones geográficas distintas y participar interactivamente (Google Docs).	Dentro de la red privada, esta característica es la misma a pesar de encontrarse en la misma ubicación geográfica, los usuarios siguen teniendo la ventaja de colaborar y participar interactivamente.

Fuente: [35], [36], [Elaboración propia]

Tabla 1.3 Ventajas de *Cloud Computing* modelo público y modelo privado

Características	Cloud Computing pública	Cloud Computing privada
Conexión Constante a Internet	Dado que todos los documentos, los programas, las aplicaciones en sí, se encuentran alojadas en la Nube, sin una conexión constante, el acceso a todos los recursos es imposible. Este aspecto es crítico sobre todo para los lugares en donde no existe un acceso fácil a Internet, ya sean áreas rurales o simplemente en la calle o en un lugar donde no ofrezcan este servicio.	En este modelo, la conexión a la red privada debe ser constante y para acceder a la Nube no es necesario contar con una conexión a Internet pues la infraestructura está ubicada dentro de la misma corporación.
No funciona bien con una conexión lenta	Se toma en cuenta que el acceso a la Nube se realiza a través de una red, el acceso a una aplicación de escritorio podría ser mucho más rápido. La velocidad de descarga debe ser alta para poder trabajar con normalidad, de lo contrario cada petición que se envíe tardaría demasiado en llegar al usuario. Es por tal razón que la conexión debe ser de alta velocidad, lo cual implica un costo mayor en la contratación del servicio de Internet.	La red privada presenta menos retardos en la conexión pues la distancia que debe recorrer cada petición es más corta, en este sentido esta debilidad presentada por el modelo público, es mejorada en el modelo privado.
Los datos en la Nube podrían no estar seguros	Este es uno de los aspectos más cuestionados de <i>Cloud Computing</i> ya que los datos se encuentran en las instalaciones de un proveedor y la capacidad de cómputo es compartida. El usuario no tiene el control sobre la seguridad ni sobre los recursos que están siendo utilizados por la Nube, es el proveedor quien se encarga de dichas cuestiones. Por tales razones, los usuarios no tienen el agrado de que alguien más tenga el control sobre lo que les pertenece.	Los datos almacenados se encuentran en la misma infraestructura y red de donde se está conectando el usuario; es decir, el usuario o la corporación, tiene control de los datos de los cuales son propietarios.
Incompatibilidad entre proveedores	Los proveedores de <i>Cloud Computing</i> no ofrecen servicios compatibles entre sí; por tal razón, si un usuario decidiera cambiar de proveedor, los datos que se encuentran almacenados en la Nube del primero, no podrían ser transportados para que almacenen en la Nube del segundo. Por ejemplo, <i>Amazon's "Simple Storage Service" [S3]</i> es incompatible con <i>IBM's Blue Cloud</i> , o <i>Google</i> , o <i>Dell</i> .	No existiría la necesidad de cambiar de proveedor, pero si se tuviera la necesidad de migrar los datos de la Nube privada a una pública, debería tomarse en cuenta la compatibilidad existente entre ambas.

Fuente: [35], [36], [Elaboración propia]

Tabla 1.4 Desventajas de *Cloud Computing* modelo público y modelo privado

1.7 RELACIÓN CON OTRAS TECNOLOGÍAS

1.7.1 GRID COMPUTING

Cloud Computing es a menudo confundida con *Grid Computing*, una forma de computación distribuida mediante la cual un gran equipo virtual se compone de un conjunto de computadores conectados en red, débilmente acoplados, y actuando en conjunto para realizar tareas, a menudo, muy grandes.

La computación *Grid* se ha utilizado en entornos en los cuales los usuarios hacen pocas pero grandes peticiones. Por ejemplo, un laboratorio puede tener mil nodos y los usuarios realizan peticiones para todos los mil, o quinientos, o doscientos, etc. así que sólo unas pocas de estas peticiones pueden ser atendidas a la vez y las otras necesitarán ser programadas para atenderse después de que los recursos sean liberados. Esto da lugar a complejos algoritmos de planificación de tareas y de procesamiento en paralelo. [28]

En la Tabla 1.5 se muestran algunos criterios comunes entre ambas tecnologías, que han sido objeto de discusión en cuanto a su semejanza.

Criterios	<i>Grid Computing</i>	<i>Cloud Computing</i>
Virtualización	En sus inicios	Esencial
Desarrollo de Aplicación	Local	En la Nube
Acceso	Vía <i>middleware</i> para <i>Grids</i>	Vía protocolos Web estándares
Organización	Virtual	Física
Modelos de Negocios	Compartidos	Pago por consumo bajo demanda (Si se trabaja con una nube pública)
SLA	Aún no exigidos	Esencial
Control	Descentralizado	Centralizado (<i>Data Center</i>)
Facilidad de Uso	Baja	Alta
Costo por cambio de infraestructura (proveedor)	Bajo, debido a la estandarización	Alto debido a incompatibilidades entre proveedores

Fuente: [27]

Tabla 1.5 Grid vs. Cloud Computing

1.7.2 WEB 2.0

La Web 2.0 fue denominada así por Dale Dougherty²² desde el año 2004. No es considerada como una nueva tecnología, sino como un fenómeno social dado que los usuarios utilizan contenido que ellos mismos generan.

“La Web 2.0 se refiere a una nueva generación de webs basadas en la creación de contenidos producidos y compartidos por los propios usuarios del portal”. [29]

En la Web 1.0, el creador del contenido lo publicaba y los usuarios accedían a ese contenido estático únicamente con la posibilidad de leerlo. La Web 2.0 permite, mediante la colaboración, mantener el contenido actualizado. Algunos ejemplos de la Web 2.0 son: *wikis*²³, blogs, Sistemas Gestores de Contenido (*Content Management Systems, CMS*²⁴), sindicación de contenidos con formato (*Really Simple Syndication, RSS*²⁵), redes sociales, *mashups*, etc.

En un futuro cercano, llegará la Web 3.0, que es la evolución de la Web 2.0, que permitirá contar con contenidos personalizados en base al uso de buscadores inteligentes que serán capaces de interpretar búsquedas personalizadas para cada usuario.

“La Web 3.0 busca construir una web con sentido semántico²⁶, interpretar, conectar, poner a disposición, datos a medida de las necesidades e intereses de cada usuario, lo que permitirá tener mejor segmentación y personalización.

²² **Dale Dougherty** de *O'Reilly Media* utilizó este término en una conferencia en la que hablaba del renacimiento y evolución de la Web.

²³ **Wiki** es un sitio web cuyas páginas web pueden ser editadas por múltiples voluntarios a través del navegador web.

²⁴ **CMS (Content Management System)** permiten facilitar la generación y edición de contenidos para la Web por parte de personal sin formación específica en programación.

²⁵ **RSS (Really Simple Syndication)** es un estándar que permite al usuario suscribirse a fuentes de información para recibir notificaciones cuando hayan nuevos contenidos o estos sean actualizados, evitando tener que visitar los sitios web frecuentemente.

²⁶ **Web Semántica** trata de convertir la información en conocimiento, ordenar y clasificar los contenidos de la web para que los ordenadores sean capaces de interpretarlos y tomar decisiones a través del cruce de datos.

Tratará de combinar contenido semántico, inteligencia artificial, inteligencia colectiva y gestión del conocimiento”. [31]

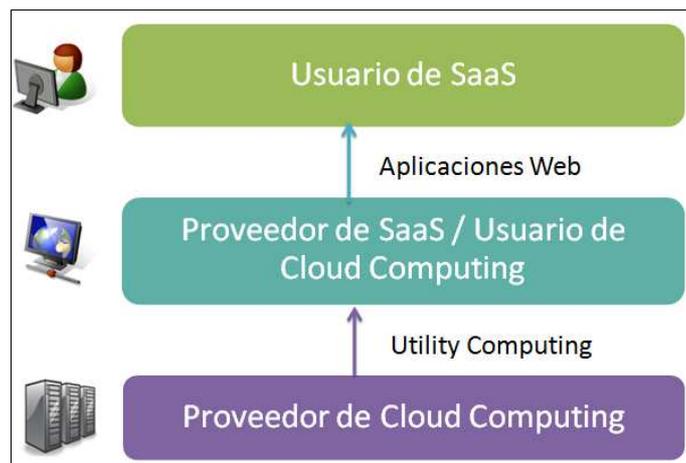
“Cloud Computing está influida por las tendencias hacia la virtualización, la automatización, el procesamiento masivamente paralelo y la orientación al servicio. Cloud Computing surge como consecuencia de las expectativas creadas por la Web 2.0 entre los usuarios. Incluso hay voces que sostienen que la idea de que una aplicación no tenga que existir en un lugar concreto, sino que pueda estar compuesta de múltiples piezas procedentes de múltiples sitios, eso se lo debemos a la Web 2.0”. [2]

Cloud Computing forma parte de esta nueva tendencia ya que es la tecnología que permite que esto se desarrolle. Como ya se había mencionado antes, *Cloud Computing* admite la colaboración entre usuarios, lo que da la pauta para entender que tanto la Web 2.0 así como la Web 3.0 abarcan esta forma de ofrecer servicios.

1.7.3 UTILITY COMPUTING

Es un modelo de suministro de recursos computacionales como un servicio. El proveedor hace que los recursos y la gestión de la infraestructura estén disponibles para los usuarios, y les cobra por el uso específico en lugar de una tarifa plana. Este modelo es similar a las utilidades públicas tradicionales como la electricidad, el agua, el gas natural o el teléfono. [32], [33]

Tiene la ventaja de tener un costo nulo o muy bajo para adquirir hardware y los recursos computacionales son alquilados. Los clientes que realizan procesamiento de datos a gran escala o que están frente a un pico de demanda también pueden evitar los atrasos que resultarían de adquirir y ensamblar físicamente una gran cantidad de computadoras.



Fuente: [34]

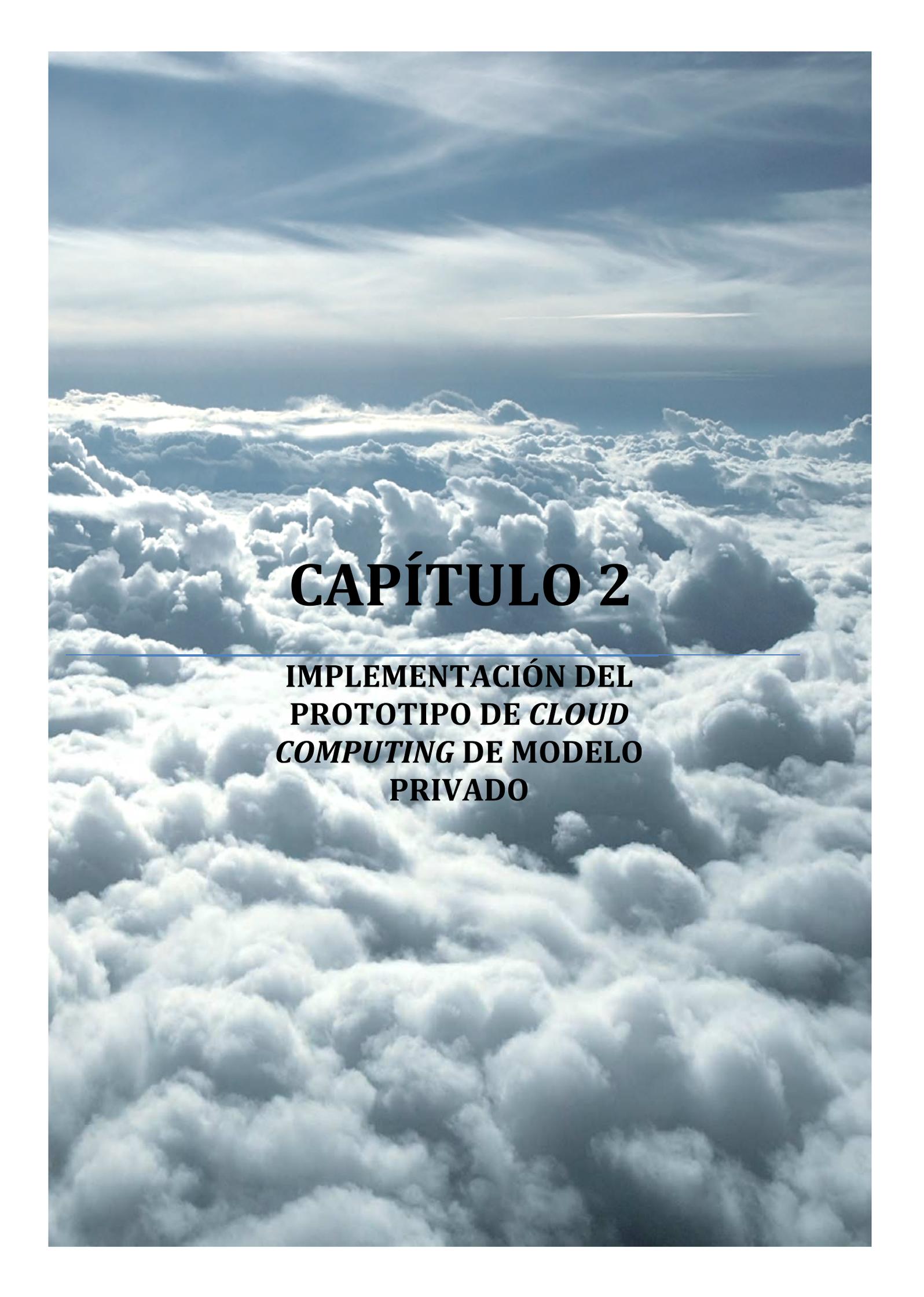
Figura 1.13 Relación entre usuarios y proveedores de *Cloud Computing*

En la Figura 1.13 se puede observar que el nivel superior puede ser recursivo; en él, los proveedores de SaaS pueden ser a la vez usuarios de SaaS. Por ejemplo, un proveedor que usa *mashups* para rentar mapas puede ser un usuario de *Craigslist*²⁷ y de los servicios de *Google maps*²⁸.

Cloud Computing se implementa sobre un conjunto organizado de computadores (*grid* o *cluster*) y utiliza el concepto de *Utility Computing* para poder ofrecer servicios bajo demanda.

²⁷ **Craigslist** es una red centralizada de comunidades urbanas en línea, ofreciendo anuncios clasificados gratis de (empleo, vivienda, sentimentales, artículos para la venta/trueque/se busca, servicios, comunidad).

²⁸ **Google maps** es un servidor de aplicaciones de mapas en la Web. Google ofrece imágenes de mapas desplazables, así como fotos satelitales del mundo entero e incluso la ruta entre diferentes ubicaciones.



CAPÍTULO 2

**IMPLEMENTACIÓN DEL
PROTOTIPO DE *CLOUD*
COMPUTING DE MODELO
PRIVADO**

2.1 UBUNTU SERVER EDITION-KARMIC KOALA (9.10)

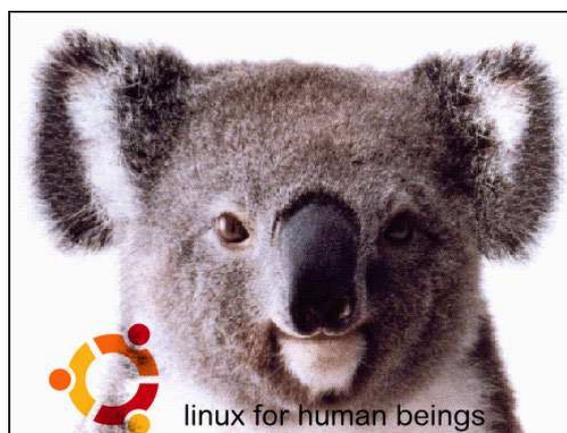
Es una distribución de Linux que no consta de interfaz gráfica, puesto que es innecesaria para el despliegue de servidores. En la Figura 2.1 se presenta el logotipo oficial de Ubuntu.



Fuente: [50]

Figura 2.1 Logotipo de Ubuntu

Esta versión de Ubuntu fue lanzada en octubre de 2009; su nombre se debe a que es la primera distribución de Linux que incluye en su CD de instalación los módulos necesarios para instalar los componentes de *Ubuntu Enterprise Cloud* (UEC), los cuales traen consigo el *middleware* para *Cloud Computing* denominado *Eucalyptus*. En la Figura 2.2 se muestra el logotipo de la distribución de Ubuntu-Karmic Koala.



Fuente: [51]

Figura 2.2 Logotipo de Ubuntu 9.10- Karmic Koala

Características:

- Soporte para sistema de archivos EXT4.

- *Kernel*²⁹ 2.6.31.
- *Grub*³⁰ 2 por defecto.

2.1.1 *UBUNTU ENTERPRISE CLOUD (UEC)*

Es un conjunto de aplicaciones desarrolladas por Canonical Ltd.³¹, incluidas en *Ubuntu Server Edition* (9.10 y 10.04) junto con un gran número de otras aplicaciones de código abierto; *Eucalyptus* es su componente principal.

UEC facilita la instalación y configuración de los componentes dentro de un entorno de *Cloud Computing* [7], [44] ya que permite instalar los paquetes correspondientes a un *front-end* y a un nodo.

2.2 *EUCALYPTUS*

El proyecto *Eucalyptus* empezó como una investigación en el Departamento de Ciencias de la Computación en la Universidad de California (Santa Bárbara) en 2007, liderado por el profesor Rich Wolski, donde se estudiaba la computación distribuida y el *hosting*³² de máquinas virtuales en grandes servidores. Una vez que el proyecto tuvo forma y se tenía definido un entorno de ejecución, empezó la integración de los AWS (*Amazon Web Services*) en su tecnología para una plataforma de Nube local, de donde nació el software de infraestructura *Eucalyptus*. [46]

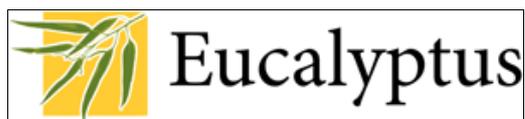
²⁹ **Kernel** es el componente central de un sistema operativo, facilita la funcionalidad de las aplicaciones, representa el puente entre los procesos de las aplicaciones y el procesamiento de datos subyacente, y maneja funciones que se realizan a nivel del hardware.

³⁰ **Grub** es un gestor de arranque. Permite a los usuarios tener múltiples sistemas operativos en la misma computadora y permite también elegir cuál ejecutar cuando la misma se inicia. [74]

³¹ **Canonical Ltd.** es una empresa privada fundada y financiada por el empresario sudafricano Mark Shuttleworth para la promoción y desarrollo de proyectos de software libre como Ubuntu. Tiene oficinas ubicadas en Londres, Montreal, Lexington, Massachusetts y Taipei.

³² **Hosting** o alojamiento web es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía Web.

Eucalyptus (*Elastic Utility Computing for Linking your Programs to Useful Systems*) es software de código abierto (*open-source*), auspiciado por *Eucalyptus Systems Inc.* (Figura 2.3), para la implementación de *Cloud Computing* (privada o híbrida), para ofrecer Infraestructura como Servicio (IaaS) utilizando *clusters*³³, los cuales proveen una interfaz que es compatible con el servicio de *Amazon EC2* (véase Sección 2.3).



Fuente: [38]

Figura 2.3 Logotipo de *Eucalyptus Systems*

Eucalyptus permite crear un entorno propio de *Cloud Computing* con el objetivo de maximizar los recursos informáticos (*clusters*, *workstations*, varios servidores y máquinas de escritorio) y proporcionar un entorno de *Cloud Computing* a sus usuarios. [43]

Eucalyptus funciona con la mayoría de distribuciones Linux, incluyendo Ubuntu, *Red Hat Enterprise Linux (RHEL)*, CentOS, *SUSE Linux Enterprise Server (SLES)*, OpenSUSE, Debian y Fedora. De la misma manera, *Eucalyptus* puede usar una gran cantidad de tecnologías de virtualización, incluyendo VMware, Xen y KVM para implementar las abstracciones que *Cloud Computing* ofrece. [37], [38]

El objetivo del proyecto *Eucalyptus* es crear una comunidad de investigación y desarrollo sobre las tecnologías de *Elastic*, *Utility* y *Cloud Computing*, para ayudar a desarrollar nuevas técnicas y estudio de modelos de uso para el *hosting*, que sea de código abierto (*open-source*) y que entre toda la comunidad, se haga una herramienta potente para estos servicios.

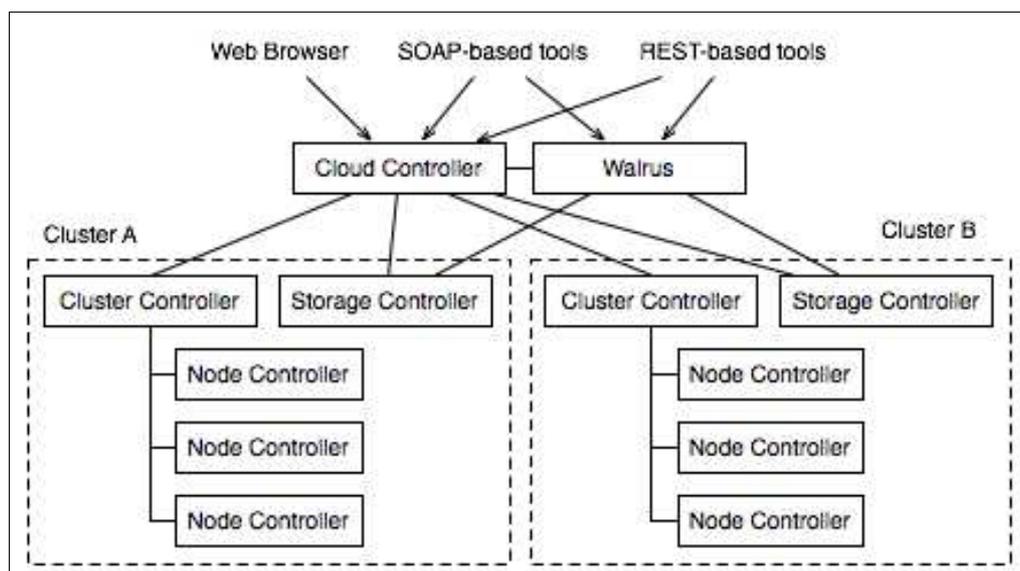
La versión más reciente de *Eucalyptus* es la 2.0; en este proyecto se utilizó *Eucalyptus v1.6.2*, ya que se consideró que es la más estable. [46]

³³ **Cluster** es un conjunto de computadores hacia los cuales se distribuye la carga (peticiones) que recibe el *Cluster Controller* (Sección 2.2.1.2)

2.2.1 ARQUITECTURA DE *EUCALYPTUS*

La arquitectura de *Eucalyptus* es simple, flexible y modular con un diseño jerárquico que consta de cinco componentes basados en Servicios Web (Figura 2.4):

- *Cloud Controller (CLC)*
- *Walrus Controller*
- *Storage Controller (SC)*
- *Cluster Controller (CC)*
- *Node Controller (NC)*



Fuente: [38]

Figura 2.4 Configuración de los componentes de *Eucalyptus*

Estos componentes pueden estar instalados en las máquinas físicas de acuerdo a cualquiera de las siguientes configuraciones: *front-end* o nodo.

Front – End

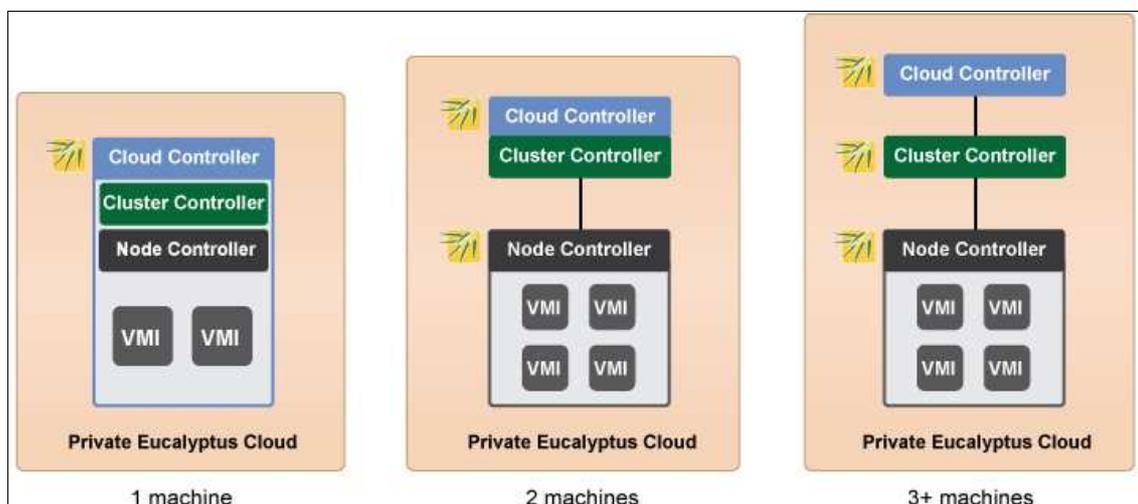
Es la máquina física que, generalmente, contiene a los elementos del *Cloud Controller (CLC)* y del *Cluster Controller (CC)*. El *front-end* es el punto de entrada

a la Nube, la cual provee una interfaz a través de la cual los servidores, la red y el almacenamiento, pueden ser evaluados y administrados. [7]

La configuración más básica para la implementación es utilizando la misma máquina física para que se ejecuten el CLC, CC, SC y *Walrus* (Figura 2.5). En una configuración más avanzada, se puede optar por ejecutar el CLC, *Walrus* (Sección 2.2.1.4), SC y CC en máquinas físicas separadas, o pueden ser combinados de acuerdo a las necesidades. Por ejemplo, una razón para separar los componentes de *Eucalyptus* puede ser mejorar el rendimiento general del sistema mediante la distribución de la carga de trabajo (es decir, *Walrus* ejecutándose en un equipo que tiene un disco duro rápido, mientras que el CLC se puede colocar en otro equipo con una CPU rápida). [38]

Nodo

Es la máquina física que contiene únicamente a los elementos del *Node Controller* (NC).



Fuente: [45]

Figura 2.5 Ejemplos de configuraciones de *Eucalyptus* en máquinas físicas

Cada componente de *Eucalyptus* expone un API en forma de un documento WSDL³⁴ que contiene las operaciones que el servicio puede realizar y la entrada y salida de estructuras de datos. La autenticación entre servicios internos se realiza a través de mecanismos estándar con *WS-Security*³⁵. Cada uno de los cinco componentes de alto nivel dispone de una interfaz propia.

Eucalyptus permite a los usuarios: iniciar, controlar, acceder y terminar máquinas virtuales utilizando interfaces web; para la comunicación interna segura, utiliza una emulación de SOAP con *WS-Security* de Amazon EC2. Es decir, los usuarios de *Eucalyptus* pueden interactuar con el sistema, utilizando las mismas herramientas e interfaces que usan para interactuar con Amazon EC2.

Actualmente, *Eucalyptus* tiene soporte para máquinas virtuales a través de los *hypervisors*: Xen, KVM/Qemu y VMware. [38], [39]

2.2.1.1 Controlador del Nodo (*Node Controller, NC*) [42]³⁶

El *Node Controller* es el componente que se ejecuta en la máquina física que alberga a las máquinas virtuales y es responsable de su puesta en marcha, inspección, terminación y limpieza. Típicamente existen varios NCs en una instalación de *Eucalyptus* y cada NC puede gestionar varias máquinas virtuales.

La interfaz del NC es descrita a través de un documento WSDL que define la estructura de datos y las operaciones de control de la instancia (*runInstance*, *describeInstance*, *terminateInstance*, *describeResource* y *startNetwork*).

³⁴ **WSDL** (*Web Services Description Language*) es un formato XML que se utiliza para describir la interfaz pública de Servicios Web y formatos de mensajes para la comunicación.

³⁵ **WS-Security** (WS-S Seguridad en Servicios Web) es un protocolo de comunicaciones que suministra un medio para aplicar seguridad a los Servicios Web.

³⁶ **[42]**: Desde la Sección 2.2.2.1 hasta la Sección 2.2.2.4 han sido realizadas tomando como referencia única la fuente número 42 de la bibliografía especificada, debido a la escasez de información acerca del tema.

runInstance describe y termina operaciones en una configuración mínima del sistema, seguido de llamadas al *hypervisor* correspondiente para controlar e inspeccionar las instancias que se están ejecutando.

Para iniciar una instancia o máquina virtual (MV), el NC realiza una copia local de los archivos de imagen de la instancia (el *kernel*, el sistema de archivos, la *ramdisk*³⁷), ya sea de un repositorio remoto de imágenes o desde la caché local, se crea un punto de terminación (*endpoint*) en la red virtual (*network overlay*), y se instruye al *hypervisor* la ejecución de la instancia. Para detener una instancia, el NC instruye al *hypervisor* la terminación de la MV, elimina el *endpoint* de la red virtual, y borra los archivos asociados con la instancia (el sistema de archivos no se conserva después de que la instancia ha sido terminada). [39]

describeResource reporta las características actuales de los recursos físicos (núcleos de CPU, memoria y capacidad de disco).

startNetwork establece una configuración para la Ethernet. (Revisar la Sección 2.2.2.5 para mayores detalles).

El NC es un Servicio Web de Apache escrito en C, recibe órdenes del *Cluster Controller* y es quien administra las instancias (máquinas virtuales) que corren sobre su hardware físico.

Cada vez que un nodo es añadido a un *cluster*, se genera un nuevo vínculo al *Cluster Controller* por cada nodo. [46]

³⁷ **Ramdisk** es una parte de la memoria que ha sido configurada para parecer, a la vista de los programas, como si fuese una unidad de disco muy rápida. No es permanente. Contiene un conjunto de drivers los cuales son cargados temporalmente en la memoria, se encarga de instruir al *kernel* la puesta en marcha apropiada del sistema de archivos cuando una máquina virtual se inicia.

2.2.1.2 Controlador del *Cluster* (*Cluster Controller, CC*)

Una colección de NCs envía reportes a un único *Cluster Controller* que, típicamente, se ejecuta en el *front-end* de un *cluster*, el cual tiene acceso tanto a la red pública, así como a la red privada.

El CC es responsable de recopilar la información de estado de su colección de NCs, de la repartición de peticiones de ejecución entrantes de máquinas virtuales a cada NC individualmente, y de la administración de la configuración de las redes públicas y privadas a las que las instancias pertenecen.

El lenguaje WSDL, que describe la interfaz del CC, es similar a la interfaz del NC, con la diferencia de que cada operación está especificada en plural, (*runInstances*, *describeInstances*, *terminateInstances*, *describeResources*).

describeInstances y *terminateInstances* controlan las operaciones de ejecución que corresponden al módulo del NC.

Cuando un CC recibe una petición del tipo *runInstances* realiza una tarea de recopilación de información para determinar que los NCs puedan albergar la nueva instancia, esto lo realiza consultando a cada NC a través de la operación *describeResource*, luego escoge al primer NC que tenga los recursos suficientes disponibles.

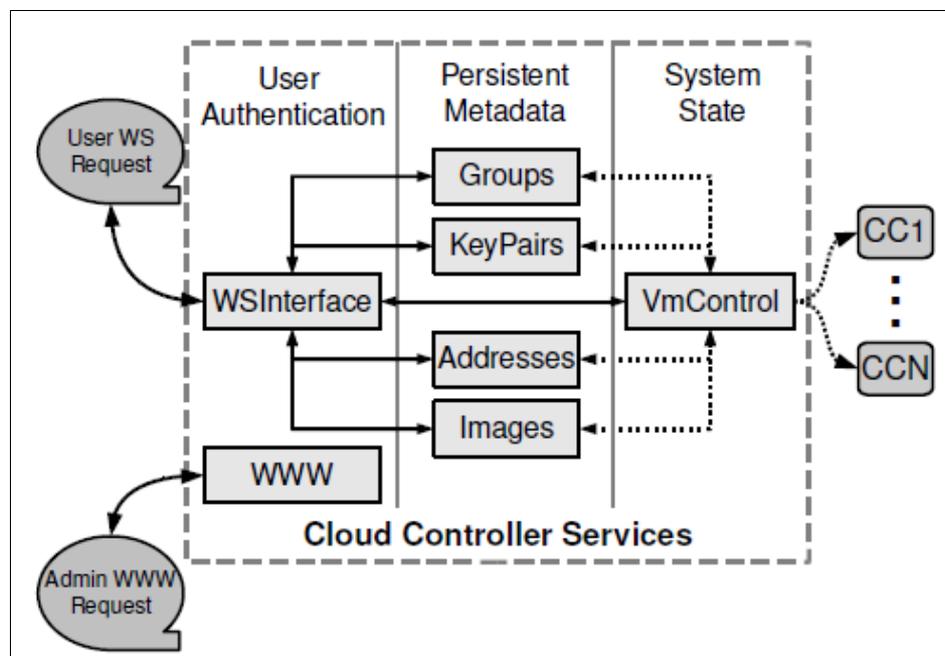
El CC también implementa la operación *describeResources*, sin embargo, en lugar de reportar los recursos físicos que están disponibles realmente, esta operación toma como entrada una descripción de los recursos que una sola instancia puede ocupar, y retorna el número de instancias de ese tipo que simultáneamente se pueden ejecutar en los NCs.

2.2.1.3 Controlador de la Nube (*Cloud Controller, CLC*)

Cada instalación de *Eucalyptus* incluye un solo *Cloud Controller*, el cual es el punto de entrada visible al usuario y el componente de decisión global.

El CLC es el responsable del procesamiento de peticiones entrantes de usuarios normales o del administrador, de la toma de decisiones de alto nivel acerca de la asignación de peticiones a las MV, del procesamiento de los SLAs y de la gestión del sistema en cuanto a persistencia y los metadatos del usuario.

El CLC está compuesto por una colección de servicios (Figura 2.6) que se encargan de las peticiones de los usuarios, de su autenticación y del mantenimiento y monitoreo de las MV. La autenticación de usuarios (*User Authentication*) presenta interfaces visibles al usuario, maneja autenticación y expone las herramientas de gestión del sistema.



Fuente: [42]

Figura 2.6 Visión general de los servicios del *Cloud Controller*

En esencia, el CLC es responsable de la gestión de los recursos subyacentes virtualizados (servidores, red y almacenamiento) a través de un API bien definido y una interfaz de usuario basada en Web. Los servicios de metadatos persistentes (*Persistent Metadata*) se encargan de la gestión de la información de recursos disponibles de imágenes y *clusters*, y de la manipulación de parámetros tales como llaves para autenticación y grupos de seguridad. Los servicios de estado de Sistema (*System State*) gestionan y monitorean los recursos de las máquinas virtuales en ejecución.

2.2.1.4 *Walrus*

Permite a los usuarios almacenar datos, organizados como *buckets*³⁸. También permite a los usuarios crear, borrar, listar *buckets* y establecer políticas de control de acceso.

La interfaz de *Walrus* es compatible con *Amazon S3*, y soporta la interfaz de administración para Imágenes de Máquinas de *Amazon* (*Amazon Machine Image, AMI*), proporcionando así un mecanismo para almacenar y acceder tanto a las imágenes de máquinas virtuales como a los datos del usuario. [41]

2.2.1.5 Controlador de Almacenamiento (*Storage Controller, SC*)

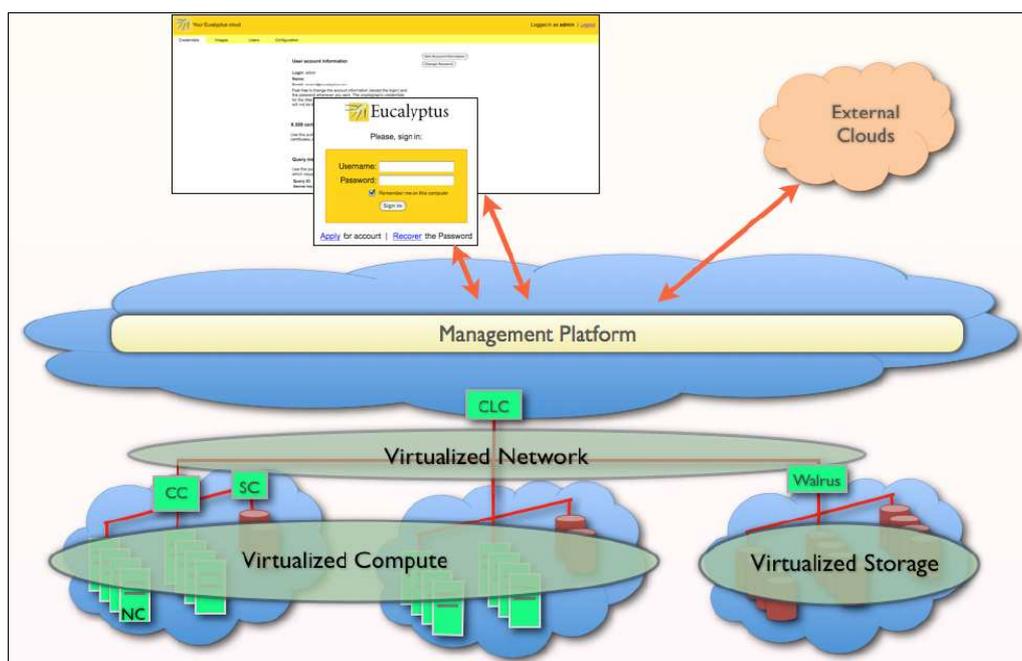
Implementa el almacenamiento de bloques para el acceso a través de la red; por ejemplo, *Amazon Elastic Block Store* (EBS). El SC es capaz de interactuar con diferentes sistemas de almacenamiento (NFS, iSCSI, etc.).

Un EBS es un dispositivo de bloque (*block device*) Linux que puede ser conectado a una máquina virtual, y envía el tráfico de disco a través de la red local a un lugar de almacenamiento remoto.

³⁸ **Bucket** es un contenedor que almacena objetos, se puede agregar o quitar objetos con los comandos *put* y *get*, hasta que la capacidad del *bucket* se agote.

Un volumen EBS no puede ser compartido con otras máquinas virtuales, pero permite la creación de un *snapshot*³⁹ en un repositorio central como *Walrus* (el servicio de almacenamiento de *Eucalyptus*) a partir del cual se puede crear un nuevo volumen que puede vincularse a una máquina virtual. [41]

2.2.2 ADMINISTRACIÓN, ACCESO Y RED VIRTUAL DE *EUCALYPTUS*



Fuente: [41]

Figura 2.7 Representación conceptual de la Nube de *Eucalyptus*

En la Figura 2.7 se puede observar la relación entre todos los componentes de una Nube basada en *Eucalyptus*. La Nube brinda recursos virtualizados al usuario y éste desde el mundo externo puede hacer uso de los mismos a través de una interfaz web o línea de comandos. El CLC es el que virtualiza los recursos subyacentes. Los CC forman parte del *front-end* para cada *cluster* definido en la Nube. Los NCs son los equipos en los cuales se ejecutan las máquinas virtuales. El SC ofrece un servicio de almacenamiento en bloque (similar a *Amazon EBS*), mientras que al sistema de almacenamiento *Walrus*, se puede acceder desde cualquier componente de la Nube. La plataforma de gestión proporciona una

³⁹ **Snapshot** es una copia total de un volumen lógico.

única consola para el administrador que le permite configurar y gestionar la Nube. La plataforma de gestión (*management platform*) también exporta varias interfaces para el administrador, desarrolladores y otros usuarios, con niveles personalizables de acceso y privilegios.

2.2.2.1 Interfaz del Cliente (*Client Interface*)

El servicio de la interfaz de cliente del CLC actúa fundamentalmente como un traductor entre las interfaces internas del sistema de *Eucalyptus* (es decir el NC y las interfaces de control de instancias del CC) y alguna de las interfaces externas del cliente. Por ejemplo, *Amazon* ofrece un documento WSDL que describe una interfaz de cliente de Servicios Web basados en SOAP, así como un documento de descripción de una interfaz HTTP basado en consultas, (cada una de las cuales pueden ser traducidas por el servicio de interfaz de usuario del CLC, en objetos internos de *Eucalyptus*).

2.2.2.2 Interfaz de Administración (*Administrative Interface*)

Además de apoyar las funciones primarias, tales como iniciar y terminar instancias, la infraestructura de una Nube debe soportar tareas administrativas, tales como añadir y eliminar usuarios e imágenes de disco. *Eucalyptus* apoya estas tareas a través de una interfaz web, implementada por el CLC, y herramientas de línea de comandos.

Los usuarios se agregan a *Eucalyptus* ya sea a través de la acción de un administrador o llenando un formulario *on-line* que se envía al administrador para su aprobación. El control sobre la creación de cuentas queda en manos de una persona, lo cual se considera necesario dada la ausencia de métodos de aprobación automatizados, tales como la verificación de tarjetas de crédito utilizados por *Amazon*.

Es el administrador el encargado de aprobar las peticiones de cuentas de usuario

y se encarga de verificar la identidad del solicitante, con la ayuda de información en el formulario de inscripción.

Actualmente, las imágenes de disco pueden ser añadidas al sistema únicamente por un administrador. Una imagen generalmente se compone del *kernel* compatible con el sistema operativo, un archivo de imagen del sistema de archivos y, opcionalmente, una imagen *ramdisk*. Además, una imagen se constituye cargando estos tres componentes en el sistema y dándole un determinado nombre. Después de que una imagen ha sido añadida, cualquier usuario puede ejecutar instancias de esa imagen. Los administradores pueden deshabilitar temporalmente o eliminar permanentemente la imagen.

Finalmente, el administrador se encarga de agregar y quitar nodos a partir de la configuración del CC.

2.2.2.3 Administración de zonas de disponibilidad (*availability zones*)

Se ha implementado un SLA inicial, que permite a un usuario observar y en base a eso elegir el nombre del *cluster* en el que va a ejecutar una instancia. Mientras los proveedores de recursos, generalmente, usan conjuntos de máquinas en términos de "*clusters*", *Eucalyptus* ha adoptado el concepto más general de "zonas", que es utilizado actualmente por *Amazon EC2*. Dentro de *EC2*, una "zona" se correlaciona con una ubicación geográfica como "la costa este de EE.UU." o "la costa oeste de EE.UU.", mientras que *Eucalyptus* usa el término para referirse a una colección lógica de máquinas que tiene varios NCs y un solo CC.

Eucalyptus permite a los usuarios especificar la configuración de zonas para la ejecución de instancias, lo que permite a un conjunto de instancias residir en un solo *cluster* o potencialmente a través de varios *clusters*. Cada configuración ofrece diferentes características administrativas y de rendimiento de la red.

En la implementación de este proyecto, el conjunto predeterminado de las zonas suministrado, permite al usuario solicitar un *cluster* específico, el menos utilizado, cualquier *cluster*, o varios *clusters*, salvo que éste o éstos no proporcionen el mínimo de recursos requerido.

2.2.2.4 Red Virtual de *Eucalyptus*

Eucalyptus ofrece una solución de red para máquinas virtuales que permite contar con conectividad, aislamiento y rendimiento.

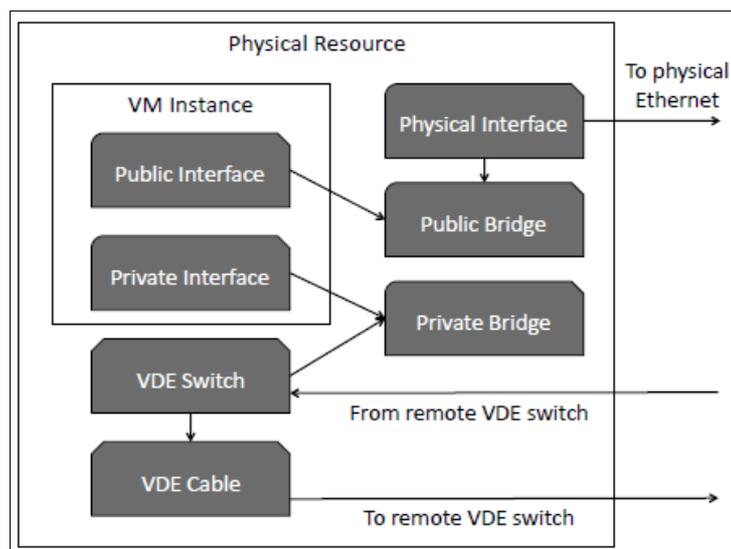
Dentro de un sistema de *Cloud Computing*, las MVs que pertenecen a un mismo *cluster*, pueden comunicarse, mientras que las que pertenecen a distintos, permanecerán aisladas. Las características que actualmente ofrecen los diferentes *hypervisors*, no soportan este tipo de diferenciación entre grupos de usuarios.

Cada instancia controlada por *Eucalyptus* tiene dos interfaces de red virtuales: la una llamada "*public*" (pública) y la otra llamada "*private*" (privada). La interfaz *public*, se encarga de mantener contacto con el mundo externo, es decir mantiene la comunicación entre la MV y un grupo de MVs o con MVs pertenecientes a una misma zona disponible (*availability zone*), como lo que se especificó en la Sección 2.2.2.3. En un entorno en el que existen varias IP públicas disponibles, éstas pueden ser asignadas a las MVs en tiempo de ejecución, permitiendo la comunicación desde y hacia la MV.

En entornos en los que las MVs están conectadas a una red privada con un *router* que soporta la comunicación externa a través de *Network Address Translation (NAT)*, a la interfaz *public* se le puede asignar una IP privada, permitiendo así la comunicación fuera de la red local a través del *router* con NAT habilitado⁴⁰. La interfaz privada asignada a la MV, es únicamente utilizada para la comunicación entre MVs a través de las zonas, incluyendo la situación en la que dos MVs

⁴⁰ Este es el entorno en el que se ha desarrollado el proyecto, dentro de la red de la Escuela Politécnica Nacional.

necesiten comunicarse y se encuentren en distintas redes privadas (zonas). La configuración de red básica de una instancia se puede observar en la Figura 2.8, la cual tiene la interfaz *public* conectada a una red pública a través de un Puente Público (*Public Bridge*), el cual está conectado directamente a la Interfaz Real (*Physical Interface*).



Fuente: [42]

Figura 2.8 Interfaces *public* y *private* de una MV de *Eucalyptus*

En *Eucalyptus*, el CC se encarga de la puesta en marcha y terminación de las interfaces de red virtuales de una instancia.

El CC puede ser configurado para adaptar la interfaz de red pública de acuerdo a los siguientes tres escenarios: la primera configuración instruye a *Eucalyptus* a adjuntar la interfaz pública virtual de la instancia a un *bridge* Ethernet basado en software, el cual está conectado a la red de la máquina física real, permitiendo al administrador de red manejar peticiones DHCP de la MV, de la misma manera que se manejan regularmente las peticiones DHCP. La segunda configuración permite al administrador definir un conjunto dinámico de direcciones IP que serán asignadas a través de un servidor DHCP, el cual es ejecutado por el CC. En esta configuración, el administrador define una red, una interfaz en el CC que está conectado a esa red, y un rango de direcciones IP que se asignan dinámicamente cuando las instancias se han iniciado. Finalmente, *Eucalyptus* soporta una

configuración que permite que un administrador defina duplas de direcciones MAC e IP. En este modo, a cada nueva instancia creada por el sistema se le asigna una dupla MAC / IP libre, la cual es liberada cuando la instancia se termina.

La interfaz privada de la instancia se conecta, mediante un *bridge*, a un sistema virtual Ethernet, basado en software, llamado *Virtual Distributed Ethernet (VDE*⁴¹, Figura 2.8). El VDE es una aplicación a nivel de proceso del protocolo Ethernet, donde los usuarios pueden especificar y controlar un *switch* Ethernet virtual y abstracciones del cable físico, que se implementan como programas que se ejecutan en el espacio del usuario. Una vez que una red VDE se ha creado, las conexiones a las redes Ethernet reales pueden ser establecidas a través de las interfaces universales TUN/TAP⁴², que, en esencia, permiten la comunicación de paquetes Ethernet desde el *kernel* de Linux, con los procesos de usuario.

Cuando *Eucalyptus* se inicia, se establece una *overlay network*⁴³ basada en VDE, que consiste de un conmutador (*switch*) VDE por cada CC y NC y tantos procesos de cables VDE como conexiones puedan establecerse entre los *switches*.

Los *switches* VDE soportan STP⁴⁴ (*spanning tree protocol*), que permite enlaces redundantes, mientras que previene bucles en la red, dando así a la red VDE un nivel de redundancia cuando los *switches* están totalmente conectados.

En tiempo de ejecución de una instancia, el NC es el responsable de controlar que la máquina virtual cree un nuevo *bridge* Ethernet, el cual será conectado al *switch* VDE local, y además configura a la instancia para adjuntar su interfaz privada al nuevo puente. En este punto, el primer requerimiento de que una instancia tenga conectividad se cumple, ya que cuando cualquier máquina virtual se inicia y está

⁴¹ Para más información acerca de VDE, consultar en: <http://vde.sourceforge.net/>.

⁴² **TUN/TAP** son drivers del núcleo de red virtual, implementan dispositivos de red en base a software. TAP implementa dispositivos Ethernet capa 2. TUN implementa dispositivos IP capa 3.

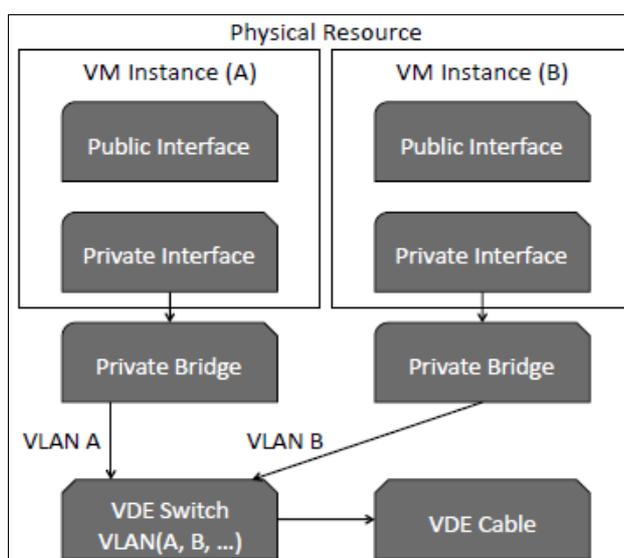
⁴³ **Overlay network** red que se construye sobre otra red.

⁴⁴ **STP** es un protocolo de la capa enlace de red que asegura una topología libre de bucles para cualquier LAN puenteadas.

conectada a un VDE, el NC será capaz de comunicarse con cualquier otra máquina virtual a través de la red Ethernet virtual, independientemente de la configuración de la red física subyacente. La versión utilizada del sistema permite al administrador, definir una subred IP de clase B, la cual va a ser utilizada por las instancias conectadas a la red privada, y a cada nueva instancia se le asignará una dirección IP dinámica dentro de la subred especificada.

El segundo requerimiento de la red virtual es que la instancia soporte aislamiento del tráfico de red. A cada conjunto de instancias, de propiedad de un usuario en particular, se le asigna una etiqueta que se utiliza como identificador de red de área local virtual (*Virtual Local Area Network, VLAN*) asignado a las instancias de ese usuario. Una vez que el identificador de VLAN ha sido asignado, todos los puertos del *switch* VDE, que están conectados a las interfaces privadas de las instancias, se configuran para etiquetar todo el tráfico entrante con el identificador de la VLAN y únicamente reenviar paquetes que tienen la misma etiqueta.

La Figura 2.9 muestra cómo dos instancias de propiedad de los usuarios A y B, ejecutándose en el mismo recurso físico, están conectadas a la red VDE a través de puertos configurados para reenviar únicamente el tráfico basado en una VLAN asignada a una MV en particular.



Fuente: [42]

Figura 2.9 Aislamiento del tráfico de dos MVs a través de etiquetas VLAN

2.3 *AMAZON WEB SERVICES (AWS)*

AWS (Figura 2.10) es una creciente unidad dentro la compañía Amazon.com que ofrece una importante variedad de soluciones de *Cloud Computing* a empresas PyMES así como a grandes organizaciones, a través de su infraestructura interna.

Amazon ofrece servicios a través de su Nube pública mediante tarificación de precios en función del tiempo de uso, ancho de banda consumido, la zona en la que se quiera ejecutar la instancia, etc. Su gran ventaja competitiva es ofrecer Infraestructura como Servicio y Plataforma como Servicio.

Para encontrar información más detallada acerca de los servicios AWS se recomienda revisar el Anexo D.



Fuente: [92]

Figura 2.10 Logo de *Amazon Web Services*

2.4 *OTRAS OPCIONES DE SISTEMA OPERATIVO Y DE MIDDLEWARE PARA CLOUD COMPUTING*

En todas las distribuciones, excepto en UEC, la instalación de *Eucalyptus*, se realiza a través de paquetes dependiendo de la distribución.

La mayoría de distribuciones de Linux que soportan *Eucalyptus* son de software libre, lo cual no representa una ventaja de UEC sobre las demás distribuciones. Sin embargo, en la Sección 2.1.1 se especificó que UEC ya trae incluidos, en la distribución, los paquetes de *Eucalyptus*; por lo tanto, permite reducir el tiempo de implementación de una Nube.

Otras opciones de *middleware* para la implementación *Cloud Computing* privada son *Enomaly*, *Open Nebula*, *IBM Blue Cloud*, tanto de software propietario como de software libre (consultar Anexo A). Actualmente *Eucalyptus* está ampliamente difundido alrededor del mundo; Eucalyptus Systems, se ha encargado de eso, y por lo tanto, existen varias fuentes de información, foros, soporte, etc.

Otra razón por la cual se optó utilizar a *Eucalyptus* en este proyecto, es que Ubuntu lo eligió para su versión basada en *Cloud Computing (Ubuntu Enterprise Cloud, UEC)*.

La NASA también utilizó *Eucalyptus* para implementar su Nube llamada NEBULA⁴⁵, la cual está conformada de otros componentes como el sistema de archivos Lustre⁴⁶, *Django* un *framework* para aplicaciones web, el motor de búsquedas SOLR, y otros.

Finalmente, *Eucalyptus* es compatible con AWS, uno de los proveedores más grandes de *Cloud Computing* en IaaS y, por lo tanto, representa una elección estratégica a la hora de implementar un sistema basado en Nube.

2.5 INSTALACIÓN Y CONFIGURACIÓN DE LOS COMPONENTES DE UBUNTU ENTERPRISE CLOUD

En esta sección se describirá la organización de los componentes físicos y la configuración e instalación de cada componente de software instalado, en base a las funciones descritas en la Sección 2.2.1.

⁴⁵ NEBULA, más información en: nebula.nasa.gov.

⁴⁶ Información tomada de: <http://www.informationweek.com/news/government/enterprise-architecture/showArticle.jhtml?articleID=217600714>.

2.5.1 REQUERIMIENTOS DE HARDWARE

De acuerdo a la configuración de los sistemas dedicados de UEC, se especifican los requerimientos para el *front-end* y cada nodo.

2.5.1.1 *Front – End*

- *Cloud Controller (CLC)*.
- *Cluster Controller (CC)*.
- *Walrus*.
- *Storage Controller (SC)*.

Hardware	Mínimo	Sugerido	Utilizado en <i>Front-End</i>	Notas
CPU (GHz)	1	2 x 2	3	Para <i>front-ends</i> donde corren varios componentes de <i>Eucalyptus</i> , es de gran ayuda utilizar un procesador, por lo menos, del tipo Dual Core ⁴⁷ .
Memoria RAM (GB)	2	4	4	La interfaz web basada en Java, necesita grandes cantidades de memoria.
Espacio en disco (GB)	40	200	200	40GB es sólo el espacio necesario para que una única imagen, caché, etc., <i>Eucalyptus</i> no trabaja sin espacio en disco.
Disco (rpm)	5400 IDE	7200 SATA	7200 SATA	Discos lentos funcionarán, pero una instancia tardará mucho más tiempo en iniciar.
Red (Mbps)	100	1000	1000	Las imágenes de MV constan de cientos de MB, y se deben copiar a través de la red a los nodos.

Fuente: [89]

Tabla 2.1 Requerimientos de hardware para un *front-end* UEC

⁴⁷ Revisar el Anexo C.

2.5.1.2 Nodo

- *Node Controller (NC)*

Hardware	Mínimo	Sugerido	Utilizado N (1,2)	Notas
CPU	Extensión de Virtualización	Extensión de Virtualización, Multicore de 64 bits	Extensión de Virtualización, Quad core de 64 bits	Equipos de 64-bits pueden ejecutar tanto en i386 como en amd64; por defecto, <i>Eucalyptus</i> sólo ejecutará una VM por núcleo de CPU en un nodo.
Memoria RAM (GB)	1	4	4	Memoria adicional, significará más y más grandes <i>guests</i> .
Espacio en disco (GB)	40	400	400	
Disco (rpm)	5400 IDE	7200 SATA o SCSI	7200 SATA	Los nodos de <i>Eucalyptus</i> utilizan el disco intensivamente; operaciones de E/S tendrán que esperar probablemente para mejorar el rendimiento en cuellos de botella.
Red (Mbps)	100	1000	1000	Las imágenes de MV constan de cientos de MB, y se deben copiar a través de la red a los nodos.

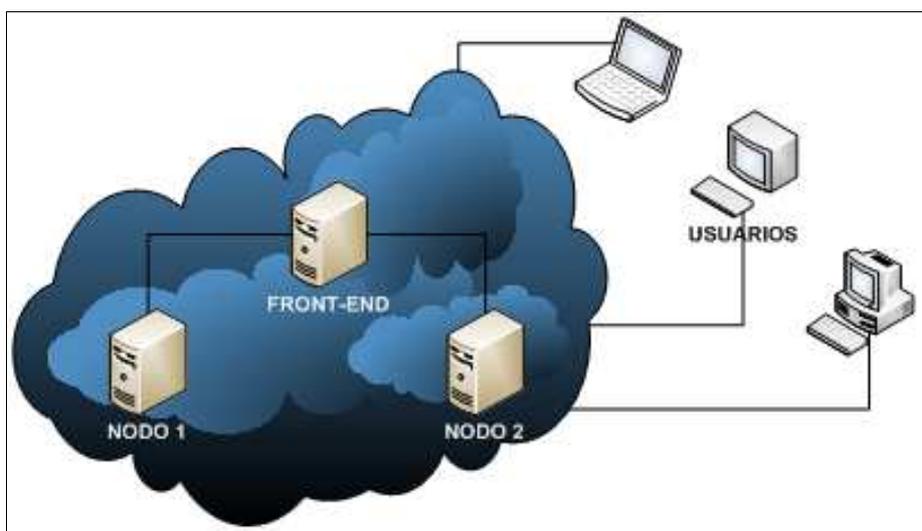
Fuente: [89]

Tabla 2.2 Requerimientos de hardware para un Nodo UEC

En el caso de contar con más equipos, se puede realizar otro tipo de configuración. Revisar Sección 2.3.1 (*Front-end*).

2.5.2 DISEÑO DE LA TOPOLOGÍA DE RED Y UBICACIÓN DE LOS COMPONENTES FÍSICOS

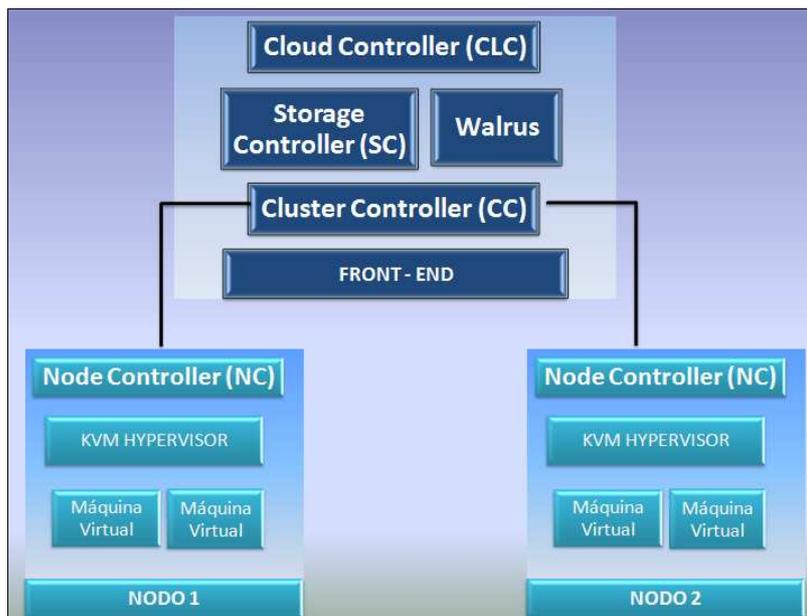
La configuración de las máquinas físicas responde a uno de los modelos básicos especificados en la Figura 2.5 de la Sección 2.2.1, utilizando dos sistemas dedicados *front-end* y nodo. El prototipo de Nube consta de tres equipos, conectados a través de un *switch* (Figura 2.11 y Figura 2.13).



Fuente: [Elaboración propia]

Figura 2.11 Configuración de los equipos físicos

Si se requiere escalar el sistema, se podría optar por agregar más NCs al componente CC de *Eucalyptus*, el cual está instalado en el *front-end*. Los componentes de *Eucalyptus* instalados en cada equipo físico se muestran en la Figura 2.12.



Fuente: Basada en [44]

Figura 2.12 Componentes de *Eucalyptus* instalados en cada equipo físico

La Tabla 2.3 muestra las direcciones IP asignadas dentro de la red de la Escuela Politécnica Nacional correspondientes a la VLAN 172.31.9.X, y la red privada de los componentes de *Eucalyptus* correspondientes a la red 10.0.0.X.

Máquina/Interfaz	eth0	eth1	eth0:priv	eth0:pub
Front-End	10.0.0.1 (asignada después de la instalación)	172.31.9.21/192.188.57.170 (asignada durante la instalación)	192.168.1.1 (asignada por <i>Eucalyptus</i>)	172.31.9.22 (asignada por <i>Eucalyptus</i>)
Nodo1 (br0)	10.0.0.2 (asignada al bridge que utiliza eth0)	X		
Nodo2 (br0)	10.0.0.3 (asignada al bridge que utiliza eth0)	X		
Administrador	172.31.9.X	X		
Usuario	172.31.9.X	X		
Máquinas Virtuales	192.168.1.X	172.31.9.22 – 172.31.9.23		

Tabla 2.3 Direcciones IP asignadas a cada máquina dentro del sistema

Adicionalmente, en la Tabla 2.3 se observa la dirección IP 192.168.1.1 asignada por *Eucalyptus* a la interfaz privada (subinterfaz⁴⁸) eth0:priv y la dirección IP 172.31.9.22 asignada a la interfaz pública eth0:pub (subinterfaz), esta última dirección es una dirección privada que en este caso es considerada como “pública”, ya que con la ayuda de *Network Address Translation* (NAT) es traducida a una dirección pública con la cual se puede tener acceso a Internet.

En la Figura 2.13 se pueden observar las redes a las que pertenecen los componentes de la Nube, los usuarios y las máquinas virtuales.

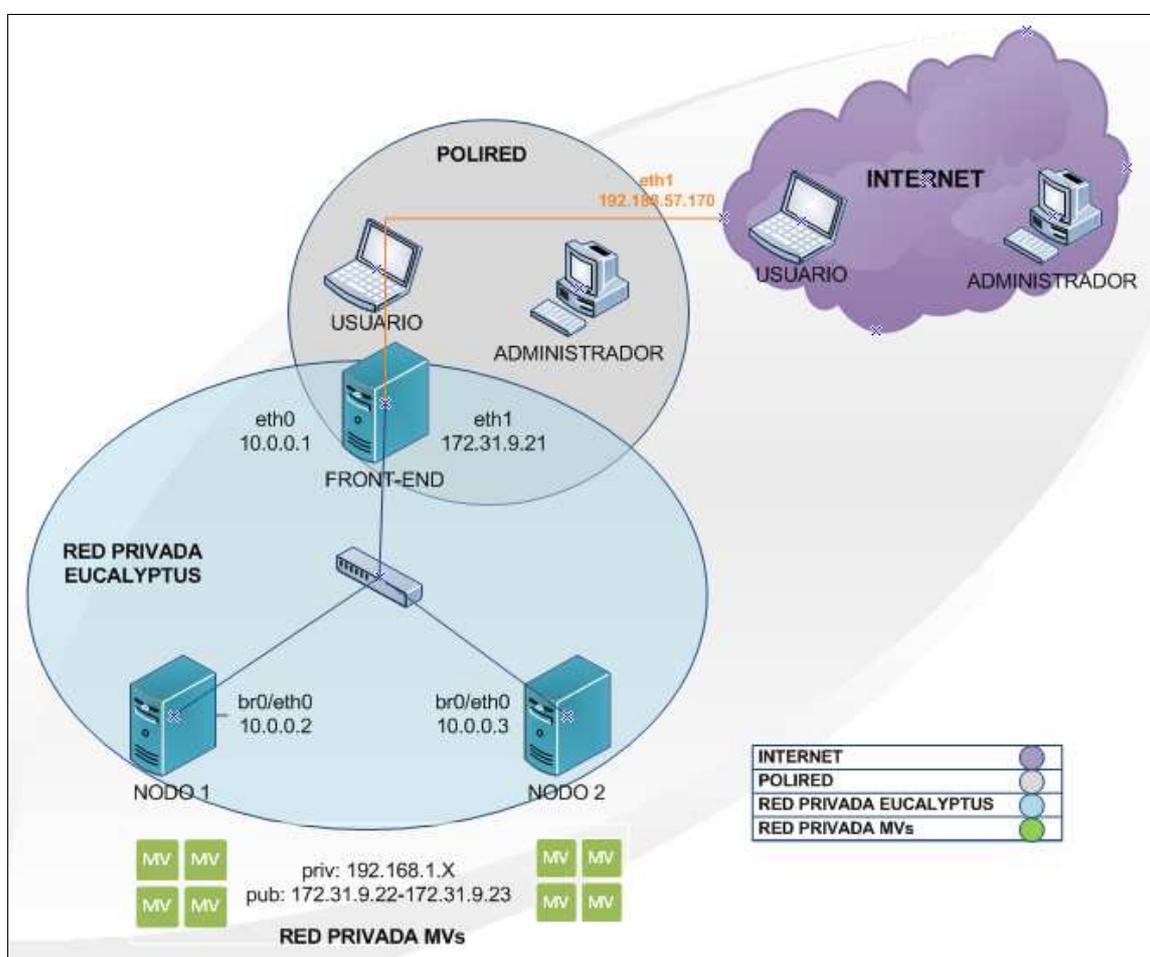


Figura 2.13 Diagrama topológico de las redes involucradas en la Nube implementada

⁴⁸ **Subinterfaz:** interfaz virtual de una interfaz física individual.

Se tiene la nube Internet desde la cual se puede acceder a los servicios de la Nube basada en *Eucalyptus*. Tanto el administrador como el usuario tienen acceso pero con diferentes permisos.

El círculo, en la parte superior de la Figura 2.13 (POLIRED), representa a la red privada de la Escuela Politécnica Nacional a través de la cual el CLC puede salir a Internet mediante NAT en la IP privada: 172.31.9.21/24 a una IP pública.

El círculo en la parte inferior de la Figura 2.13 (RED PRIVADA EUCALYPTUS), representa a la red privada de la Nube basada en *Eucalyptus* (10.0.0.X/24), a través de la cual sus tres componentes (*front-end* y dos nodos) se comunican.

Las máquinas virtuales (MVs) se encuentran en la parte inferior de la Figura 2.13, éstas se ejecutan en otra red privada (RED PRIVADA MVs) a través de la cual se pueden comunicar entre ellas y entre máquinas de otros *clusters* de la misma Nube; las máquinas virtuales pueden tener dos interfaces de red, una pública (pub) y una privada (priv) (Revisar Sección 2.2.2.4).

2.5.3 INSTALACIÓN DE *UBUNTU ENTERPRISE CLOUD 9.10*

El CD de instalación de Ubuntu server 9.10 *Karmic Koala*, se puede encontrar en: <http://releases.ubuntu.com/9.10/>⁴⁹.

En el proyecto se ha utilizado la versión de 64 bits *server install CD* para los nodos y la versión de 32 bits para el *front-end*.

2.5.3.1 *Front-End*

⁴⁹ Última visita: octubre 2010.



Figura 2.14 Instalación de *Ubuntu Enterprise Cloud*

En la Figura 2.14 se muestra la pantalla de inicio para la instalación de UEC, incluida en el mismo CD de Ubuntu Server. Se escoge la opción *Install Ubuntu Enterprise Cloud*.

En el caso de contar con un sistema ya instalado como Ubuntu Server, Centos u otra distribución de Linux compatible con *Eucalyptus*, es conveniente instalar los paquetes que *Eucalyptus Systems* Inc. ofrece en su página web: <http://open.eucalyptus.com/downloads>⁵⁰.

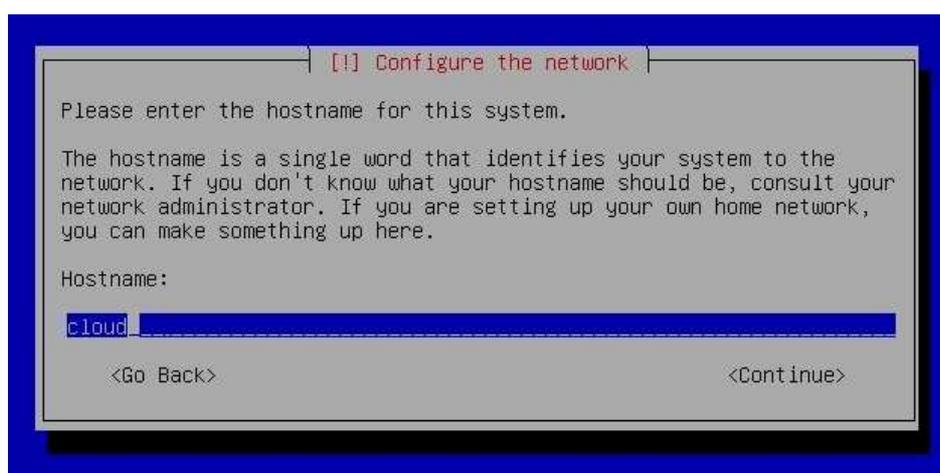


Figura 2.15 Configuración del *hostname* de la máquina

⁵⁰ Última visita: octubre 2010.

En la Figura 2.15 se muestra el *hostname* del *front-end*, el cual fue configurado como “cloud” dentro del dominio de la Escuela Politécnica Nacional (epn.edu.ec).

El instalador de UEC detecta automáticamente si existe o no un *Cluster Controller* en la red a la que está conectada la máquina donde éste se está ejecutando, si no existe, la opción correcta es escoger “Cluster” tal y como se muestra en la Figura 2.16. Esta opción instalará los paquetes correspondientes al CLC, CC, *Walrus* y SC.



Figura 2.16 Modo de Instalación de UEC

En cuanto al particionamiento del disco para el *front-end*, no se consideró necesario utilizar LVM⁵¹ debido a que se va a utilizar todo el disco para la instalación del sistema. La opción “*Guided – use entire disk*” es la más conveniente.

A continuación, se crea una cuenta de usuario (Figura 2.17) que no será utilizada para tareas administrativas; la contraseña que se escribirá en el siguiente paso será de utilidad para asignar una contraseña al usuario *root* cuando el sistema ya esté completamente instalado. Después, el programa de instalación preguntará si se desea encriptar el directorio */home*, para lo cual es aconsejable escoger la opción “S” para contar con un mayor nivel de seguridad.

⁵¹LVM (*Logical Volume Manager*) consultar en: http://es.wikipedia.org/wiki/Logical_Volume_Manager.

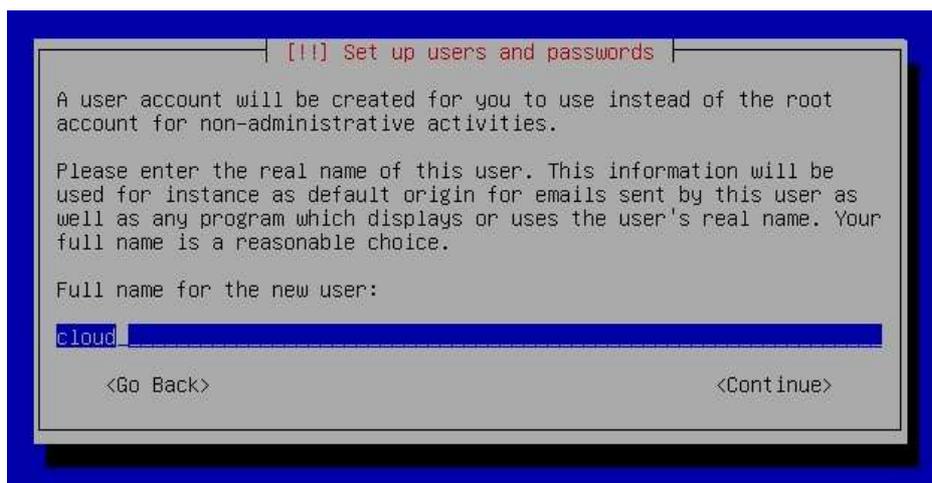


Figura 2.17 Creación de una nueva cuenta de usuario

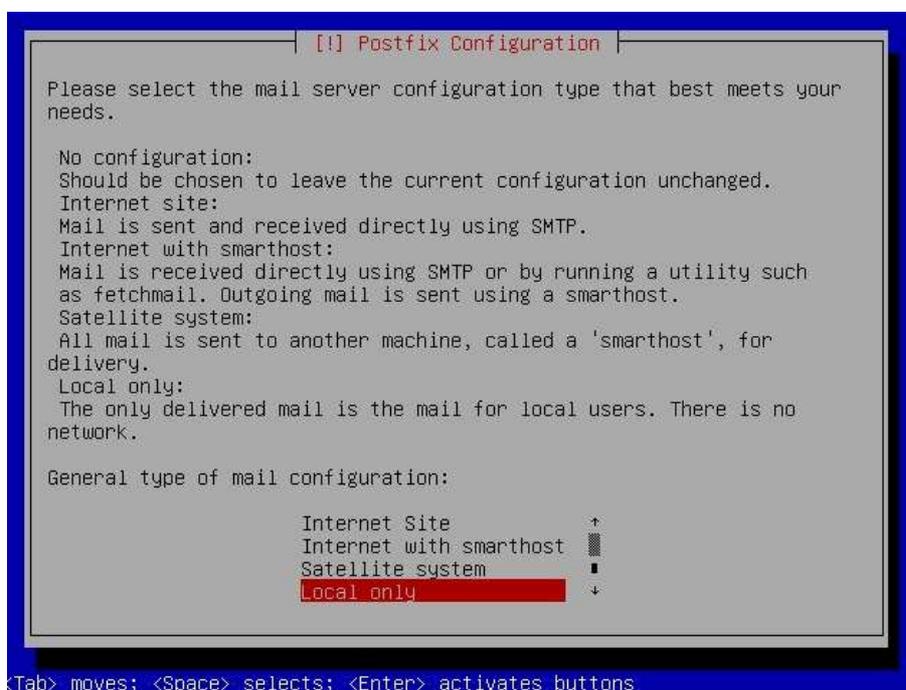


Figura 2.18 Configuración del servidor de correo electrónico (Postfix)

Al momento de escoger el tipo general de configuración del correo, “Local only” es la opción más apropiada tomando en cuenta que los correos de notificación y confirmación se van a enviar a usuarios locales⁵² (Figura 2.18). Sin embargo, sí es posible utilizar otro tipo de configuración en el caso de que se quiera enviar correo a servidores ubicados en Internet.

⁵² Políticas de la red de la EPN.



Figura 2.19 Configuración del FQDN para el correo electrónico

En la Figura 2.19 se escogió "cloud.epn.edu.ec" como el FQDN para el servidor de correo electrónico, el cual está especificado en el DNS de la Escuela Politécnica Nacional.

Una Nube puede conformarse de varios *clusters*, por eso es importante que cada uno de ellos tenga un nombre que permita identificarlos fácilmente (Figura 2.20).

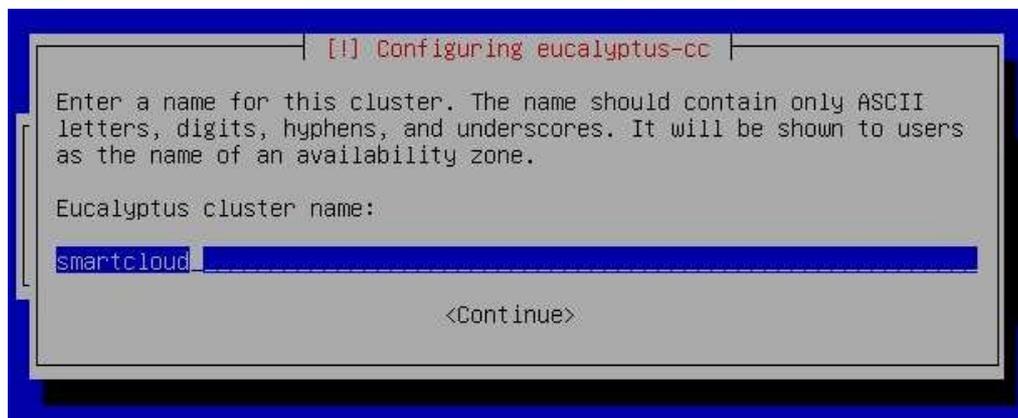


Figura 2.20 Configuración del nombre del *cluster* dentro de la Nube

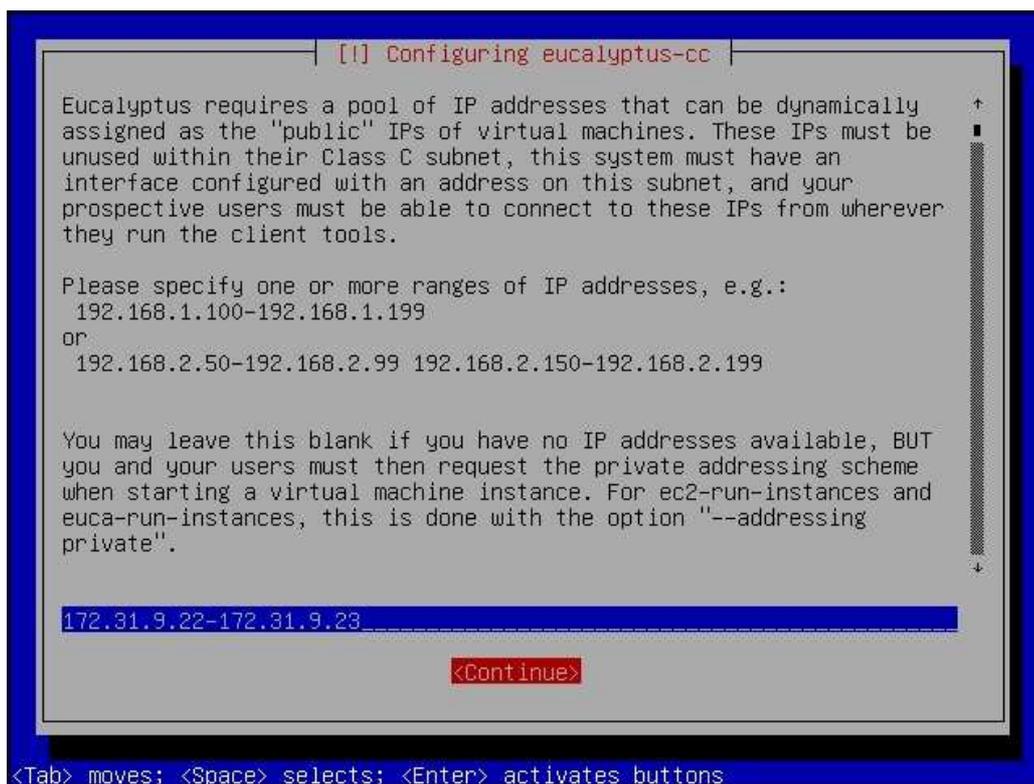


Figura 2.21 Configuración de un conjunto de IPs “públicas”

En la Figura 2.21 se especifica el grupo de direcciones IP que serán asignadas dinámicamente a las máquinas virtuales que vayan a tener acceso al mundo exterior (interfaz *public*, Sección 2.2.2.4), en este caso son IPs estáticas dentro de la red privada de la EPN.

2.5.3.2 Nodo

Ya que el *front-end* ha sido instalado, el siguiente paso es la instalación del nodo; sin embargo, no es indispensable instalar primero el *front-end*, ya que es el usuario quien escoge en las opciones cuál de los dos desea instalar, con la única diferencia de que si se instala primero el *front-end*, el instalador del nodo lo detectará, entonces se establecerá automáticamente la instalación para el mismo.

Al igual que con el *front-end*, se debe configurar un nombre y una cuenta de usuario para el nodo.

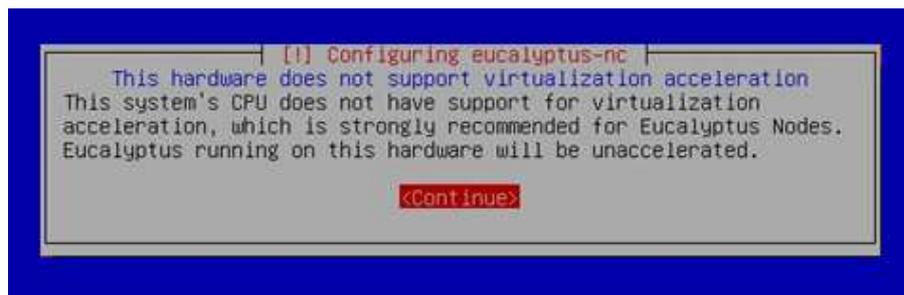


Figura 2.22 Advertencia del no uso de hardware con extensión de virtualización

En los nodos es recomendable utilizar hardware que soporte extensión de virtualización (Intel-VT⁵³ o AMD-V⁵⁴), ya que son los que van a ejecutar las máquinas virtuales, y el uso del *hypervisor* KVM⁵⁵, el cual viene por defecto instalado en Ubuntu server 9.10; se requiere de este tipo de hardware para funcionar a través de la virtualización completa (Revisar Anexo B.2.2). La Figura 2.22 muestra la advertencia cuando el hardware utilizado no soporta extensión de virtualización.

Para activar esta característica del procesador, es necesario configurarla en el BIOS ya que por defecto viene desactivada. Para mayor información revisar el Anexo C.

Cuando la instalación termina, el sistema está pre-configurado para continuar con la configuración de los servicios y de la red, donde las máquinas virtuales se van a ejecutar.

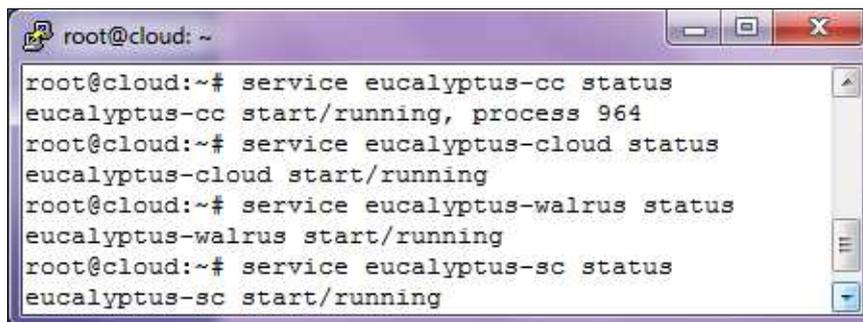
2.5.4 CONFIGURACIÓN DE LOS COMPONENTES DE *EUCALYPTUS*

Después de la instalación de UEC, los servicios de todos los componentes del *front-end* y de los nodos se deben estar ejecutando (Figuras 2.23 y 2.24).

⁵³ Intel-VT especificado en mayor detalle en el Anexo B.3.1.

⁵⁴ AMD-V especificado en mayor detalle en el Anexo B.3.2.

⁵⁵ KVM especificado en mayor detalle en el Anexo B.4.2.

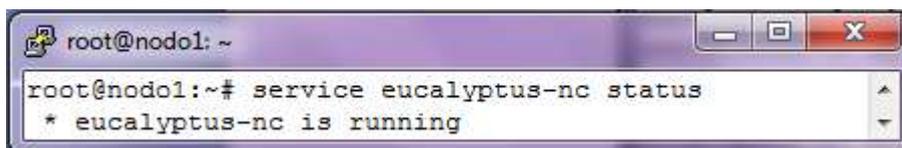


```

root@cloud: ~
root@cloud:~# service eucalyptus-cc status
eucalyptus-cc start/running, process 964
root@cloud:~# service eucalyptus-cloud status
eucalyptus-cloud start/running
root@cloud:~# service eucalyptus-walrus status
eucalyptus-walrus start/running
root@cloud:~# service eucalyptus-sc status
eucalyptus-sc start/running

```

Figura 2.23 Servicios de *Eucalyptus* activos en el *front-end*



```

root@nod01: ~
root@nod01:~# service eucalyptus-nc status
* eucalyptus-nc is running

```

Figura 2.24 Servicios de *Eucalyptus* activos en el nodo 1

El usuario “eucalyptus” del *Cloud Controller* requiere del acceso al *Walrus*, *Storage Controller* y *Cluster Controller* a través de SSH.

En los nodos es necesario configurar un *password* temporal para el usuario “eucalyptus”, mientras se copia la llave pública del *front-end* en cada nodo para tener acceso con SSH (Código 2.1), sin necesidad de estar introduciendo cada vez el *password*. Posteriormente no hace falta el *password* temporal en los nodos y por lo tanto hay que borrarlo.

```

root@grid:~#su -u eucalyptus ssh-copy-id -i
~eucalyptus/.ssh/id_rsa.pub eucalyptus@10.0.0.4

```

Código 2.1 Copia de ssh-id en el nodo 1

Después, se deben asignar las correspondientes direcciones IP a las interfaces del *front-end*.

En el Código 2.2 y en el Código 2.3, se muestran los archivos de configuración de las interfaces: `/etc/network/interfaces` correspondientes al *front-end* y al *nodo1*.

```

# Archivo de configuracion interfaces de red front-end
auto lo

```

```

iface lo inet loopback

# interfaz de comunicacion con la red privada de la EPN
auto eth1
iface eth1 inet static
    address 172.31.9.21
    netmask 255.255.255.0
    gateway 172.31.9.1

# interfaz de comunicacion del frontend con los nodos
auto eth0
iface eth0 inet static
    address 10.0.0.1
    netmask 255.255.255.0

```

Código 2.2 Archivo de configuración de las interfaces de red del *front-end*

```

# Interfaz de loopback
auto lo
iface lo inet loopback

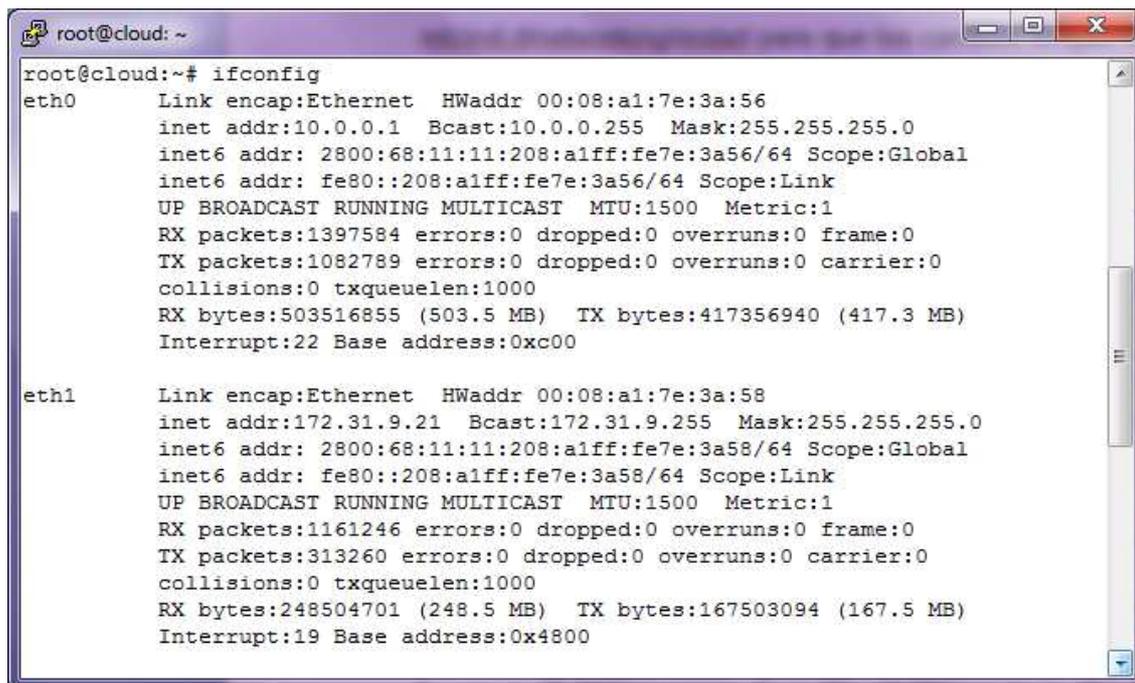
# Interfaz de red primaria
auto eth0
iface eth0 inet manual

#bridge
auto br0
iface br0 inet static
    address 10.0.0.2
    netmask 255.255.255.0
    gateway 10.0.0.1
bridge_ports eth0
bridge_maxwait 0

```

Código 2.3 Archivo de configuración de las interfaces de red del nodo1

A continuación se debe reiniciar el servicio, como usuario root: `/etc/init.d/networking restart` para que los cambios tengan efecto. En la Figura 2.25, ejecutando el comando “ifconfig” en el *front-end*, se pueden apreciar los cambios efectuados en el archivo de configuración de las interfaces.



```

root@cloud:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:08:a1:7e:3a:56
          inet addr:10.0.0.1  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: 2800:68:11:11:208:a1ff:fe7e:3a56/64 Scope:Global
          inet6 addr: fe80::208:a1ff:fe7e:3a56/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1397584 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1082789 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:503516855 (503.5 MB)  TX bytes:417356940 (417.3 MB)
          Interrupt:22 Base address:0xc00

eth1      Link encap:Ethernet  HWaddr 00:08:a1:7e:3a:58
          inet addr:172.31.9.21  Bcast:172.31.9.255  Mask:255.255.255.0
          inet6 addr: 2800:68:11:11:208:a1ff:fe7e:3a58/64 Scope:Global
          inet6 addr: fe80::208:a1ff:fe7e:3a58/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1161246 errors:0 dropped:0 overruns:0 frame:0
          TX packets:313260 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:248504701 (248.5 MB)  TX bytes:167503094 (167.5 MB)
          Interrupt:19 Base address:0x4800

```

Figura 2.25 Asignación de direcciones IP a cada interfaz en el *front-end*

En los nodos, el proceso es similar, pero ya que estos están bajo el control del *Node Controller*, la configuración de la IP de la interfaz se realiza en el bridge en el archivo `/etc/network/interfaces` de cada nodo.

En la Figura 2.26, ejecutando el comando “ifconfig” en uno de los nodos, se pueden apreciar los cambios efectuados en el archivo de configuración de las interfaces. El bridge `br0` es el que se ha configurado con la IP correspondiente al nodo.

```

root@nod01:~# ifconfig
br0      Link encap:Ethernet  HWaddr 00:1c:c0:b9:6e:a2
         inet addr:10.0.0.2  Bcast:10.0.0.255  Mask:255.255.255.0
         inet6 addr: 2800:68:11:11:21c:c0ff:feb9:6ea2/64 Scope:Global
         inet6 addr: fe80::21c:c0ff:feb9:6ea2/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:3297829  errors:0  dropped:0  overruns:0  frame:0
         TX packets:1721940  errors:0  dropped:0  overruns:0  carrier:0
         collisions:0 txqueuelen:0
         RX bytes:3641486738 (3.6 GB)  TX bytes:285127258 (285.1 MB)

eth0     Link encap:Ethernet  HWaddr 00:1c:c0:b9:6e:a2
         inet6 addr: fe80::21c:c0ff:feb9:6ea2/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:3304295  errors:0  dropped:0  overruns:0  frame:0
         TX packets:1751144  errors:0  dropped:0  overruns:0  carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:3689108484 (3.6 GB)  TX bytes:287197155 (287.1 MB)
         Memory:e0300000-e0320000

```

Figura 2.26 Asignación de direcciones IP a cada interfaz en el nodo1

Después de haber realizado la configuración de las direcciones IP en cada interfaz, el siguiente paso es registrar los componentes (*Walrus*, *Cluster*, *Storage Controller* y *Nodos*) con el *Cloud Controller* para que pueda conocer su ubicación en la red y pueda establecer la conexión con ellos. Este proceso se realiza de la siguiente manera (Código 2.4):

```

root@grid:~#euca_conf --register-walrus 172.31.9.21
root@grid:~#euca_conf --register-cluster smartcloud 172.31.9.21
root@grid:~#euca_conf --register-sc smartcloud 172.31.9.21
root@grid:~#euca_conf --register-nodes 10.0.0.2 10.0.0.3

```

Código 2.4 Registro de los componentes de *Eucalyptus*

El nombre que se le asignó al *cluster* es “smartcloud”, lo cual debe constar en el comando de registro del *Cluster* y del *Storage Controller*. Los nodos también son registrados únicamente con sus correspondientes IPs.

En el directorio: `/var/log/eucalyptus` se encuentran los archivos de log de cada componente; los archivos de log de registro tienen el sufijo: `-registration.log`; por ejemplo, el archivo de log de registro del *Cluster Controller* es `cc-registration.log`. Estos permiten determinar si existe algún inconveniente con el registro.

Para empezar a interactuar con el *Cloud Controller* es importante descargar las credenciales, las cuales permitirán el acceso de un determinado usuario. Las credenciales que el usuario necesita son de dos tipos: los certificados X.509⁵⁶ y las credenciales de interfaz de consulta (*query interface credentials*).

Para descargar y utilizar las credenciales del administrador se utiliza un script llamado `credentialscript.sh` (Código 2.5).

```
#credentials script
rm -rf ~/.euca/credenciales.zip
rm -rf ~/.euca/*.pem
rm -rf ~/.euca/eucarc
mkdir -p ~/.euca
chmod 700 ~/.euca
cd ~/.euca
euca_conf --get-credentials credenciales.zip
unzip credenciales.zip
ln -s ~/.euca/eucarc ~/.eucarc
```

Fuente: [89]

Código 2.5 Script `credentialscript.sh`

El script permite borrar, en el caso de que ya existan, el archivo `credenciales.zip`, cualquier archivo con extensión `.pem` y el archivo `eucarc`. Después se crea el directorio `/.euca` dentro del directorio `/home` utilizando la opción `mkdir -p`, se le da permisos de lectura, escritura y ejecución al propietario, y utilizando el comando `euca_conf`⁵⁷ con la opción `--get-credentials`, se guardan en el archivo `credenciales.zip`, se descomprime el archivo y se crea un *link* simbólico (`-s`) del *path* completo del archivo (`~/.euca/eucarc`) al archivo (`~/.eucarc`).

⁵⁶ **Certificados X.509** es un estándar UIT-T para infraestructuras de claves públicas (*Public Key Infrastructure* o PKI). X.509 especifica, formatos estándar para certificados de claves públicas y un algoritmo de validación de la ruta de certificación.

⁵⁷ **euca_conf** es un script que viene con *Eucalyptus* y que permite realizar la configuración del archivo `eucalyptus.conf`.

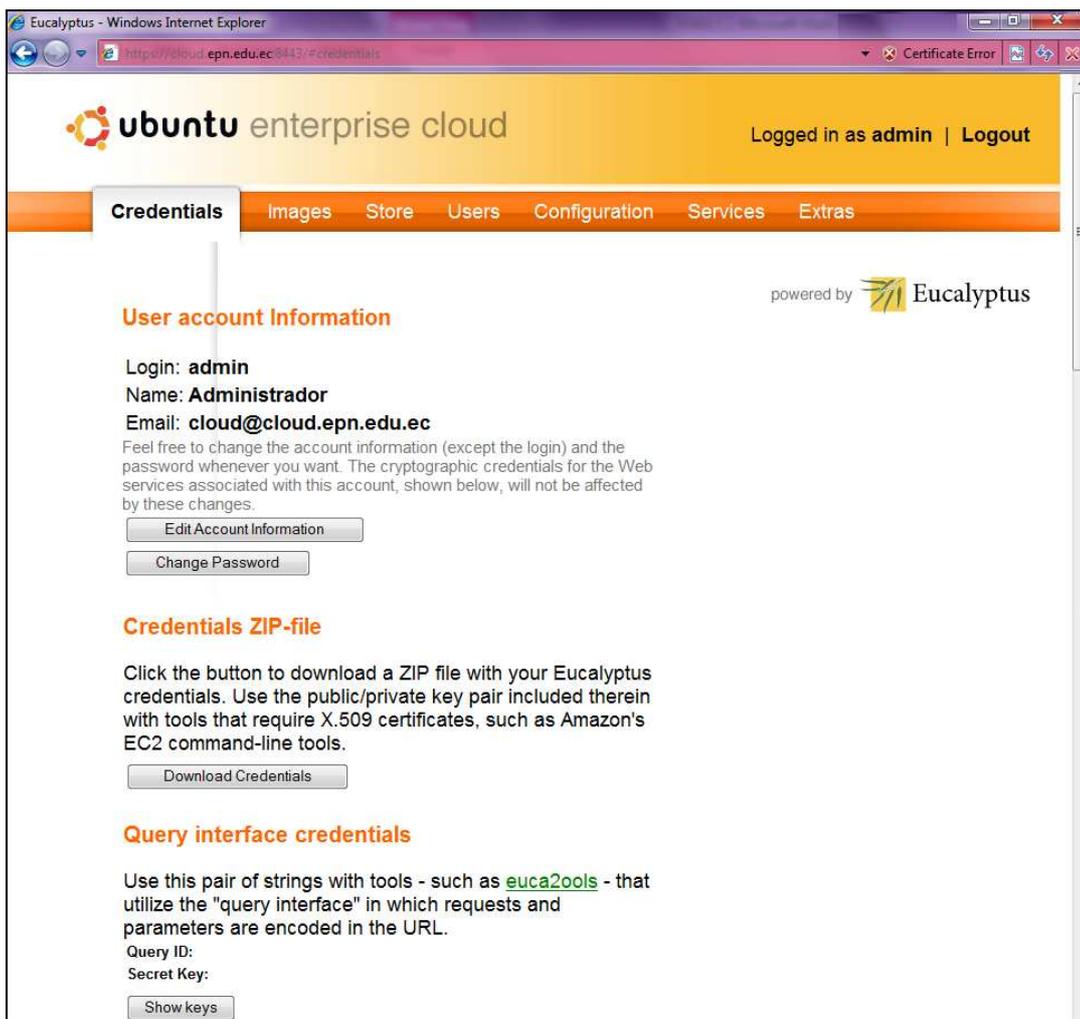


Figura 2.27 Descarga de credenciales desde interfaz web

La otra opción es utilizar la interfaz web con la URL: <https://cloud.epn.edu.ec:8443> e ingresando con la clave del administrador (admin); si es la primera vez se utiliza la clave por defecto "admin" y luego el sistema pedirá que se cambie de contraseña. A través de esta interfaz web (Figura 2.27), se puede descargar el directorio (.zip) guardarlo y descomprimirlo. Este directorio contiene los certificados especificados en el archivo eucarc (EC2_PRIVATE_KEY, EC2_CERT, EUCALYPTUS_CERT) necesarios para autenticarse con el CLC y poder utilizar las herramientas de línea de comandos (*euca2ools*, Sección 2.5.4.1).

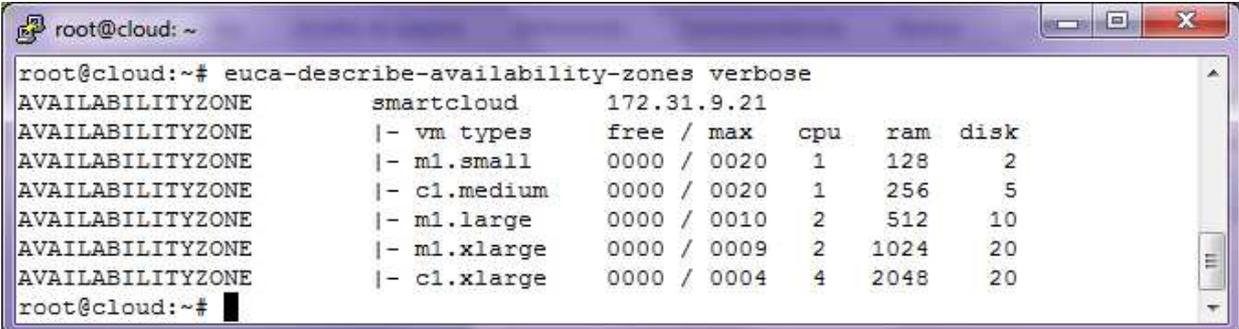
En el Código 2.6 se pueden observar las variables que *Eucalyptus* utiliza, el comando: (. ~/.euca/eucarc) o (source ~/.euca/eucarc) permite establecer estas

Entre las funcionalidades *euca2ools* se pueden destacar:

- Consulta de *availability zones*.
- Administración de llaves SSH .
- Administración de MVs.
- Administración de grupos de seguridad (*security groups*).
- Administración de volúmenes lógicos y *snapshots*.
- Administración de imágenes.
- Administración de direcciones IP.

La instalación de las *euca2ools* se realiza a través del comando “apt-get install euca2ools”.

Una vez que los componentes de la Nube han sido registrados, a través de las *euca2ools*, se puede dar el primer paso para comprobar que todos los componentes se están comunicando.



```

root@cloud: ~
root@cloud:~# euca-describe-availability-zones verbose
AVAILABILITYZONE      smartcloud      172.31.9.21
AVAILABILITYZONE      |- vm types    free / max    cpu    ram    disk
AVAILABILITYZONE      |- m1.small    0000 / 0020  1     128   2
AVAILABILITYZONE      |- c1.medium   0000 / 0020  1     256   5
AVAILABILITYZONE      |- m1.large    0000 / 0010  2     512  10
AVAILABILITYZONE      |- m1.xlarge   0000 / 0009  2    1024  20
AVAILABILITYZONE      |- c1.xlarge   0000 / 0004  4    2048  20
root@cloud:~#

```

Figura 2.28 Availability zones en smartcloud

El uso de las *euca2ools* es más sencillo gracias a las variables que se exportaron a través del archivo *euca2oolsrc*. Si no se establecen estas variables que *Eucalyptus* utiliza, cada vez que se ejecute un comando de las *euca2ools*, se deberá especificar con las opciones `-k` (EC2_PRIVATE_KEY), `--cert` (EUCALYPTUS_CERT), `--user` (EC2_USER), etc. Todas las variables y parámetros establecidos en el archivo.

Las herramientas *euca2ools* de *Eucalyptus* tienen su equivalente en *Amazon* denominadas *ec2ools*. Revisar Anexo D.1 para mayor detalle de las *ec2ools*.

En la Figura 2.28 se puede observar el número máximo de máquinas virtuales disponibles en el sistema de acuerdo a los requerimientos de cpu, ram y disco, dependiendo del tamaño de máquina virtual que se desee, los tamaños disponibles son: m1.small, c1.medium, m1.large, x1.large y c1.xlarge.

2.5.4.2 Archivos de configuración de *fron-end* y nodos

En el archivo de configuración de *Eucalyptus*: `/etc/eucalyptus/eucalyptus.conf` del Código 2.7, se asignan los siguientes valores a cada variable.

EUCALYPTUS: Representa el directorio donde *Eucalyptus* será instalado.

EUCA_USER: El nombre de usuario con el que trabaja *Eucalyptus*.

DISABLE_DNS: Establece la funcionalidad de DNS dinámico de *Eucalyptus*.

ENABLE_WS_SECURITY: Activa o desactiva el método para la comunicación segura *WS-Security* entre los componentes de *Eucalyptus*.

LOGLEVEL: Esta variable controla el nivel de la salida de los archivos de log. Las opciones son en orden descendente: 'DEBUG, INFO, WARN, ERROR, y FATAL'. Por defecto se establece en DEBUG.

CC_PORT: El puerto en el que el servicio del *Cluster Controller* está escuchando.

SCHEDPOLICY: Establece las políticas que el *Cluster Controller* utiliza para asignar a un nodo la ejecución de las MVs. GREEDY (el primer nodo que el CC encuentre será escogido para ejecutar la MV), ROUNDROBIN (los nodos son seleccionados uno después de otro), POWERSAVE (los nodos son puestos en hibernación cuando no están ejecutando MVs, y son reanudados cuando nuevos recursos son requeridos).

POWER_IDLETHRESH: Es el número de segundos que un nodo puede permanecer inactivo (es decir, sin correr máquinas virtuales) antes de que se intente apagarlo.

POWER_WAKETHRESH: Es el número de segundos que *Eucalyptus* debe esperar después de intentar despertar a un nodo.

NODES: Las direcciones IP de los nodos que son parte del *cluster*.

NC_SERVICE: El nombre del servicio del *Node Controller*.

NC_PORT: El número de puerto en el que el servicio del *Node Controller* estará escuchando.

HYPERVISOR: El *hypervisor* con el que el *Cluster Controller* estará interactuando para la ejecución de máquinas virtuales.

INSTANCE_PATH: Es el directorio que va a ser utilizado por el *Cluster Controller*, para almacenar imágenes de las instancias que se están ejecutando.

VNET_PUBINTERFACE: La interfaz de red que va a estar conectada a la red externa.

VNET_PRIVINTERFACE: La interfaz de red que va a estar conectada a la misma red que los nodos.

VNET_DHCPDAEMON: La ubicación del binario del servidor DHCP para las máquinas virtuales.

VNET_DHCPUSER: Es el nombre de usuario que utiliza el servidor DHCP de las máquinas virtuales, en el caso de que se utilice otro servidor DHCP, el nombre de usuario (diferente de *root*) se debe especificar en esta variable.

```

EUCALYPTUS="/ "
EUCA_USER="eucalyptus"
DISABLE_DNS="Y"
ENABLE_WS_SECURITY="Y"
LOGLEVEL="DEBUG"
CC_PORT="8774"
SCHEDPOLICY="ROUNDROBIN"
POWER_IDLETHRESH="300"
POWER_WAKETHRESH="300"
NODES=" 10.0.0.2 10.0.0.3"
NC_SERVICE="axis2/services/EucalyptusNC"
NC_PORT="8775"
HYPERVISOR="not_configured"
INSTANCE_PATH="not_configured"
VNET_PUBINTERFACE="eth1"
VNET_PRIVINTERFACE="eth0"
VNET_DHCPDAEMON="/usr/sbin/dhcpd3"
VNET_DHCPUSER="dhcpd"
VNET_MODE="MANAGED-NOVLAN"
VNET_SUBNET="192.168.1.0"
VNET_NETMASK="255.255.255.0"
VNET_DNS="172.31.4.2"
VNET_LOCALIP="10.0.0.1"
VNET_ADDRSPERNET="16"
VNET_PUBLICIPS="172.31.9.22-172.31.9.23"
VNET_INTERFACE="eth0"

```

Código 2.7 Archivo de configuración de *Eucalyptus* en el *front-end*

En el Código 2.8 se muestra el archivo `/etc/eucalyptus/eucalyptus.conf` del nodo 1, el del nodo 2 es similar, con la diferencia de que en el primero no se ha establecido la variable `MAX_CORES` para que tome su valor por defecto, cuatro.

```

EUCALYPTUS="/ "
EUCA_USER="eucalyptus"
DISABLE_DNS="Y"
ENABLE_WS_SECURITY="Y"
LOGLEVEL="DEBUG"
CC_PORT="8774"

```

```
SCHEDPOLICY="ROUNDROBIN"
POWER_IDLETHRESH="300"
POWER_WAKETHRESH="300"
MAX_CORES="16"
NC_SERVICE="axis2/services/EucalyptusNC"
NC_PORT="8775"
HYPERVISOR="kvm"
INSTANCE_PATH="/var/lib/eucalyptus/instances"
VNET_PUBINTERFACE="eth0"
VNET_PRIVINTERFACE="eth0"
VNET_BRIDGE="br0"
VNET_DHCPDAEMON="/usr/sbin/dhcpd3"
VNET_DHCPUSE="dhcpd"
VNET_MODE="MANAGED-NOVLAN"
```

Código 2.8 Archivo de configuración de *Eucalyptus* en el nodo 1

MAX_CORES: El número máximo de CPU / núcleos que se le permiten utilizar al NC. Por defecto se utilizan todos los núcleos disponibles.

En el nodo 2 del prototipo, se estableció esta variable en su valor por defecto considerando que cada nodo tiene 4 núcleos, en el nodo 1 se la configuró con el valor de 16. En total se pueden ejecutar máximo 20 máquinas virtuales (Figura 2.28); sin embargo ésta variable puede ser configurada con un valor mayor de acuerdo a los requerimientos que se tengan, pero hay que considerar que el uso de CPU puede incrementarse considerablemente, por tal razón sería recomendable realizar una medición de rendimiento para determinar si es conveniente aumentar el valor de esta variable o incrementar el número de nodos.

Los valores de dicha variable en cada nodo se establecieron tomando en cuenta que se iban a realizar pruebas posteriores, acerca del uso de CPU, especificadas en la Sección 3.7 y de esta manera comparar la disminución de rendimiento al ejecutar en el nodo 1 un mayor número de máquinas virtuales que en el nodo 2.

VNET_MODE: Uno de los modos de la Sección 2.5.3.2.

VNET_SUBNET: La subred que va a ser utilizada para asignar direcciones IP en las máquinas virtuales.

VNET_NETMASK: La máscara de red utilizada para la dirección de VNET_SUBNET.

VNET_DNS: Dirección IP del servidor de dominio de nombres.

VNET_LOCALIP: La dirección de la interfaz física del *front-end* conectada con los nodos.

VNET_ADDRSPERNET: Es utilizada para controlar el número de direcciones IP disponibles para cada usuario. Para escoger el valor correcto de este parámetro, es necesario tomar en cuenta el número de IPs disponibles de acuerdo a VNET_SUBNET y VNET_NETMASK, cuántas VLANs simultáneas soporta la red, y cuántas redes de usuario activas, el administrador desea que el sistema soporte.

VNET_PUBLICIPS: Esta variable determina el rango de direcciones “públicas” que serán asignadas a las máquinas virtuales, cuando éstas se ejecuten, considerando que cada una de ellas puede tener una IP del rango privado en su interfaz “privada” y una pública (Revisar la Sección 2.2.2.4).

VNET_INTERFACE: La interfaz (bridge br0) que está conectada a la interfaz de la red que tiene acceso a los nodos, está en eth0: 10.0.0.2.

HYPervisor: El *hypervisor* a través del cual se ejecutarán las máquinas virtuales.

INSTANCE_PATH: La ruta donde se guardará la información de las máquinas virtuales que se ejecutan en ese nodo.

VNET_BRIDGE: El nombre del bridge configurado por el *hypervisor* correspondiente.

2.5.4.3 Modos de Red de *Eucalyptus*

Eucalyptus puede soportar una gran variedad de características de infraestructura de red y topologías, es por eso que maneja cuatro diferentes modos de red, cada uno correspondiente a un nivel de seguridad diferente permitiendo a los administradores del sistema acoplar cada configuración de Nube de acuerdo con las políticas y manejo de las necesidades del cliente. [75]

2.5.4.3.1 *Modo System*

Es el modo de red más simple que soporta *Eucalyptus*, ya que satisface un pequeño número de especificaciones de red. En este modo, antes de que arranque una máquina virtual, *Eucalyptus* asigna una dirección MAC y una dirección IP aleatoria, y éstas se acoplan a la red Ethernet física a través del puente local (*bridge* br0) del nodo. De forma general, las máquinas virtuales obtienen una dirección IP usando DHCP, es por eso que el administrador de la red debe configurar un servidor DHCP que cuente con las direcciones IP necesarias para que puedan arrancar las máquinas virtuales que maneja. [76]

2.5.4.3.2 *Modo Static*

Este modo ofrece al administrador un mayor control sobre la asignación de direcciones IP a las máquinas virtuales.

“Cuando una máquina virtual se ejecuta, Eucalyptus crea una entrada estática dentro del archivo de configuración del servidor DHCP (/var/run/eucalyptus/net/euca-dhcp.conf), toma el siguiente par de direcciones MAC e IP, lo asigna a la máquina virtual, y acopla el dispositivo de red virtual a la red Ethernet física a través del bridge configurado en los nodos”. [76]

Una característica especial de este modo es que es utilizado por administradores que por lo general asignan las mismas direcciones a las mismas máquinas virtuales.

2.5.4.3.3 *Modo Managed*

Maneja mejores características que los dos modos anteriores. El administrador define una red de gran tamaño, generalmente privada, de la cual las máquinas virtuales obtendrán sus direcciones IP. *“Al igual que en el modo static, Eucalyptus maneja un servidor DHCP para la asignación de direcciones IP para cada máquina virtual nueva”*. [76]

Este modo presenta mayor seguridad que los modos especificados anteriormente, ya que se pueden definir grupos de seguridad y aplicar diferentes reglas de entrada a los mismos, y así se controla el tipo de tráfico que se va a transmitir. Esto es semejante a los grupos de seguridad⁶¹ que maneja *Amazon*. [76]

“El administrador puede especificar un conjunto de direcciones IP públicas que los usuarios pueden utilizar, para luego asignarlas a máquinas virtuales ya sea en el arranque o dinámicamente en tiempo de ejecución. Esta característica es similar al servicio elastic IPs de Amazon⁶²”. [76]

2.5.4.3.4 *Modo Managed-NoVLAN*

Este modo tiene las mismas características que el modo *managed*, con respecto a grupos de seguridad y asignación dinámica de direcciones IP, pero las máquinas virtuales no están aisladas unas de otras.

En el prototipo se utilizó el modo *Managed-NoVlan* debido a que proporciona varias funcionalidades al igual que el modo *Managed* excepto el aislamiento de

⁶¹ Revisar Sección 2.3.

⁶² Revisar Sección 2.3.

máquinas virtuales en la capa 2 (enlace de datos, modelo ISO/OSI). No se implementó el modo *Managed* ya que el aislamiento se realiza a través de etiquetamiento de VLANs y el *switch* con el que actualmente se han realizado las conexiones entre los equipos (*front-end* y nodos) no soporta esta tecnología. Sin embargo hay que tomar en cuenta que si se requiere aislar a las máquinas virtuales se lo puede al nivel de la capa de red (por ejemplo *subnetting*); en este caso no se lo consideró conveniente ya que la Nube implementada es del tipo privada y todas las máquinas virtuales ejecutadas en ella se pueden comunicar a través de la misma red (RED PRIVADA MVSs, Figura 2.13), siempre y cuando el usuario así lo configure en su respectivo Grupo de seguridad (Sección 2.5.7).

2.5.5 PROCESO DE INSERCIÓN DE IMÁGENES EN *EUCALYPTUS*

Una Imagen de Máquina de *Eucalyptus* (*Eucalyptus Machine Image, EMI*) es una combinación de una o varias imágenes de disco virtual, de *kernel* e imágenes de *ramdisk*, así como los tres archivos XML que contienen *metadata* acerca de la misma. Estas imágenes residen en el *Walrus* y son usadas como plantillas para crear instancias (Máquinas Virtuales) en UEC.

Por ejemplo, cada EMI de sistema operativo Linux es una combinación de un archivo XML con un nombre como “.img.manifest.xml” con información sobre una o más imágenes de disco duro, una imagen del *kernel* y una imagen *ramdisk*, cuyo identificador tiene el siguiente formato: emi-XXXXXXXX. Un archivo XML con un nombre como “vmlinuz-X.X.XX-XX-server.manifest.xml” con información sobre la imagen del *kernel* correspondiente con un identificador: eki-XXXXXXXX, y por último un archivo XML con un nombre como “initrd.img-X.X.XX-XX-server.manifest.xml” con información acerca de la *ramdisk* correspondiente, cuyo identificador es: eri-XXXXXXXX. [44]

Existen varios métodos para insertar imágenes en el sistema, a través de la interfaz web y a través de línea de comandos (con imágenes certificadas de *Eucalyptus* y a partir de un archivo .iso de un determinado sistema operativo).

Ubuntu Enterprise Cloud ofrece a través de su interfaz web, imágenes que por defecto el cliente puede utilizar, simplemente debe instalarlas, manejando el primer método que es el más sencillo; en la Figura 2.29 se puede observar que la imagen de Ubuntu 9.10 se encuentra instalada y existen más imágenes disponibles que son accesibles desde <https://imagestore.canonical.com>.

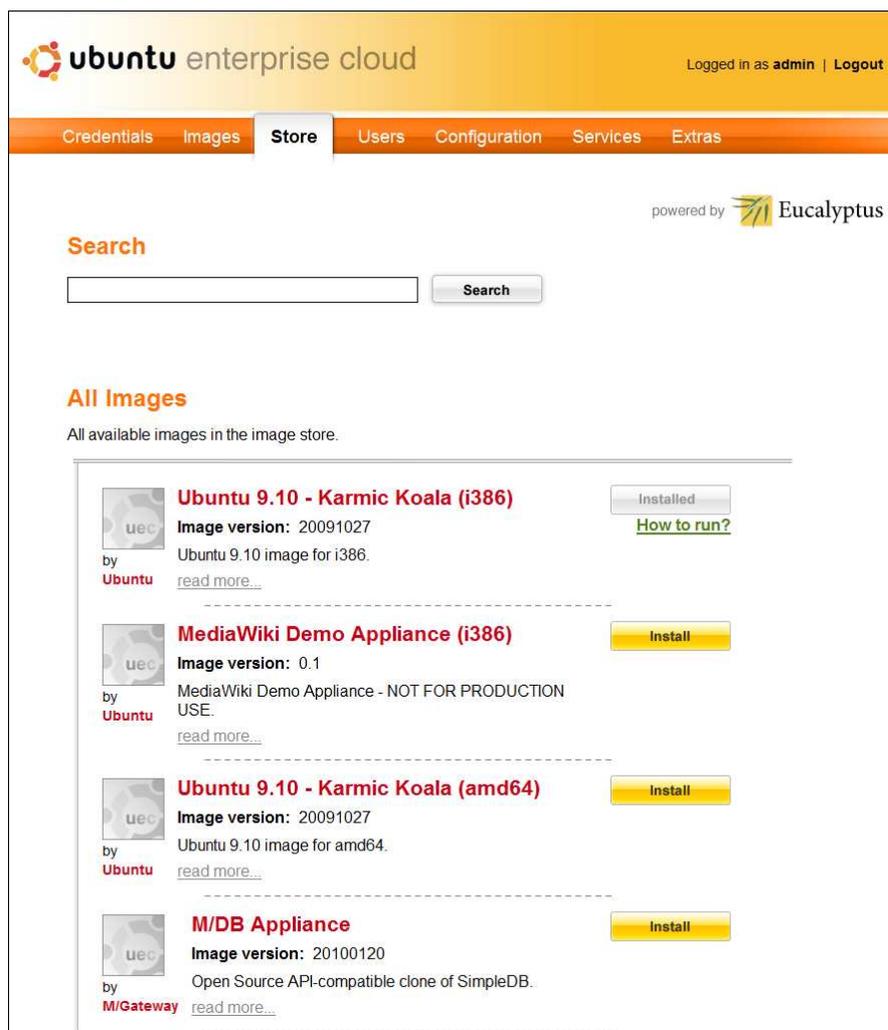


Figura 2.29 Imágenes disponibles en el Store de UEC

Para realizar la inserción de imágenes a través de las *euca2ools* (línea de comandos), se insertó una imagen del sistema operativo Centos 5.3.

En la página web oficial de *Eucalyptus*, se encuentra el enlace de descarga: <http://open.eucalyptus.com/sites/all/modules/pubdlcnt/pubdlcnt.php?file=http://www.eucalyptussoftware.com/downloads/eucalyptus-images/euca-centos-5.3-i386.tar.gz&nid=1210>, y utilizando el comando `wget` se descarga el archivo

comprimido euca-centos-5.3-i386.tar.gz, el cual puede ser descomprimido en cualquier directorio.

```
euca-bundle-image -i ~/euca-centos-5.3-i386/kvm-kernel/vmlinuz-2.6.28-11-server --kernel true

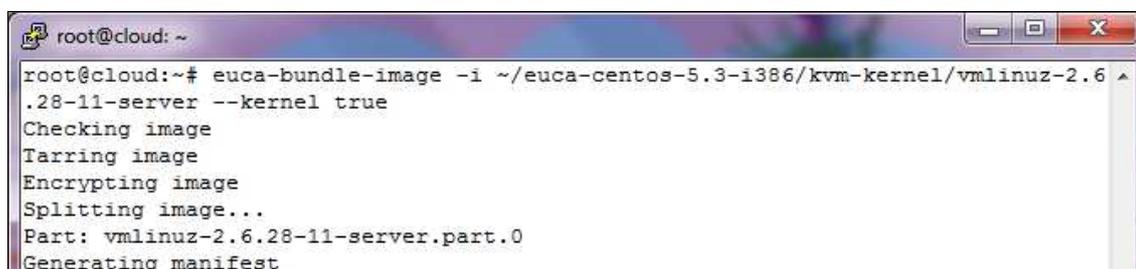
euca-upload-bundle -b centos-kernel-bucket -m /tmp/vmlinuz-2.6.28-11-server.manifest.xml

euca-register centos-kernel-bucket/vmlinuz-2.6.28-11-server.manifest.xml
```

Código 2.9 Imagen Centos (*kernel*)

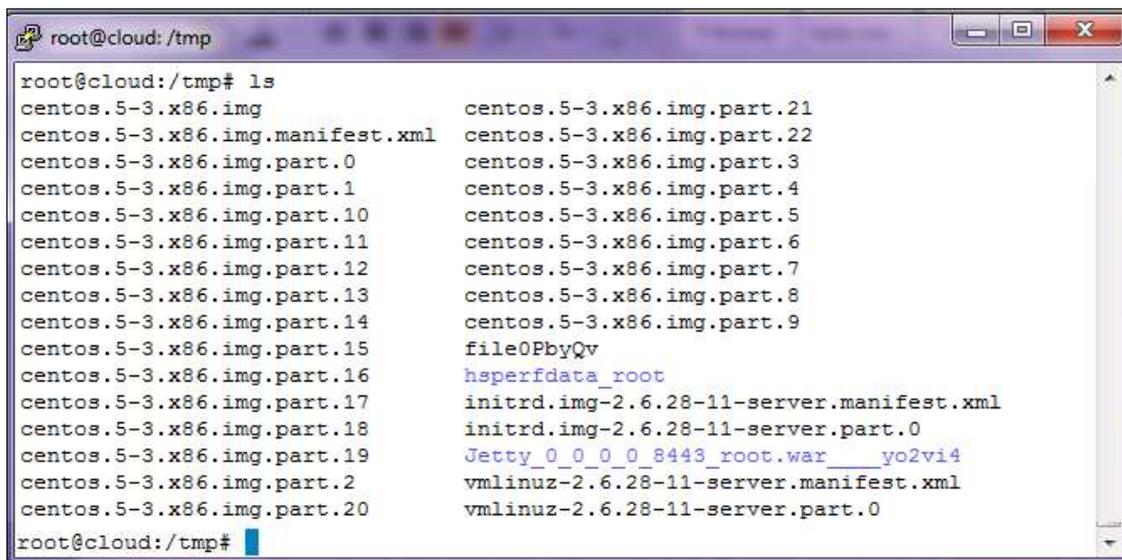
En el directorio euca-centos-5.3-i386 están dos directorios kvm-kernel y xen-kernel, lo que implica que esta imagen puede ser insertada utilizando cualquiera de los dos *hypervisors*. El Código 2.9 muestra la inserción del *kernel de la Imagen de Centos*. El comando euca-bundle-image permite “preparar” la imagen que se va a insertar, separándola en múltiples partes y generando un archivo manifest.xml (Figura 2.30), el cual contiene los metadatos acerca de la imagen; estas partes son almacenadas en el directorio /tmp (Figura 2.31) hasta que sean cargadas en *Walrus* a través del comando euca-upload-bundle (Figura 2.32).

El comando euca-register permite registrar el *kernel* insertado con *Eucalyptus*.



```
root@cloud: ~
root@cloud:~# euca-bundle-image -i ~/euca-centos-5.3-i386/kvm-kernel/vmlinuz-2.6.28-11-server --kernel true
Checking image
Tarring image
Encrypting image
Splitting image...
Part: vmlinuz-2.6.28-11-server.part.0
Generating manifest
```

Figura 2.30 euca-bundle-image (*kernel*)

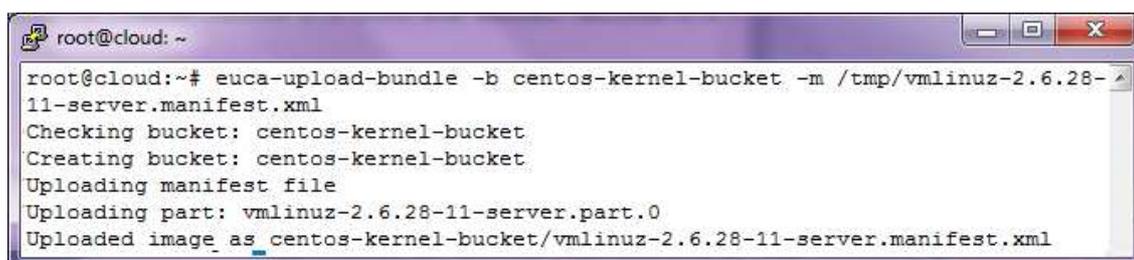


```

root@cloud: /tmp
root@cloud: /tmp# ls
centos.5-3.x86.img                centos.5-3.x86.img.part.21
centos.5-3.x86.img.manifest.xml   centos.5-3.x86.img.part.22
centos.5-3.x86.img.part.0         centos.5-3.x86.img.part.3
centos.5-3.x86.img.part.1         centos.5-3.x86.img.part.4
centos.5-3.x86.img.part.10        centos.5-3.x86.img.part.5
centos.5-3.x86.img.part.11        centos.5-3.x86.img.part.6
centos.5-3.x86.img.part.12        centos.5-3.x86.img.part.7
centos.5-3.x86.img.part.13        centos.5-3.x86.img.part.8
centos.5-3.x86.img.part.14        centos.5-3.x86.img.part.9
centos.5-3.x86.img.part.15        file0PbyQv
centos.5-3.x86.img.part.16        hspferfdata_root
centos.5-3.x86.img.part.17        initrd.img-2.6.28-11-server.manifest.xml
centos.5-3.x86.img.part.18        initrd.img-2.6.28-11-server.part.0
centos.5-3.x86.img.part.19        Jetty_0_0_0_8443_root.war_yo2vi4
centos.5-3.x86.img.part.2         vmlinuz-2.6.28-11-server.manifest.xml
centos.5-3.x86.img.part.20        vmlinuz-2.6.28-11-server.part.0
root@cloud: /tmp#

```

Figura 2.31 Partes de la imagen en /tmp

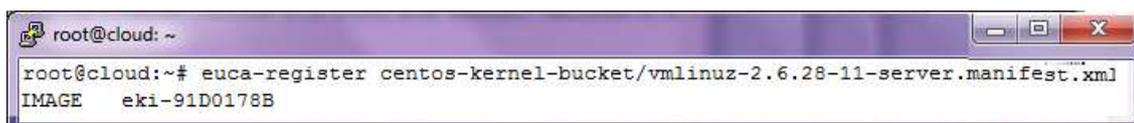


```

root@cloud: ~
root@cloud:~# euca-upload-bundle -b centos-kernel-bucket -m /tmp/vmlinuz-2.6.28-11-server.manifest.xml
Checking bucket: centos-kernel-bucket
Creating bucket: centos-kernel-bucket
Uploading manifest file
Uploading part: vmlinuz-2.6.28-11-server.part.0
Uploaded image as centos-kernel-bucket/vmlinuz-2.6.28-11-server.manifest.xml

```

Figura 2.32 euca-upload-bundle (kernel)



```

root@cloud: ~
root@cloud:~# euca-register centos-kernel-bucket/vmlinuz-2.6.28-11-server.manifest.xml
IMAGE     eki-91D0178B

```

Figura 2.33 euca-register (kernel)

Una vez que se ha registrado el *kernel* con su respectivo *bucket*, aparecerá el identificador de la imagen del *kernel*: eki-91D0178B (Figura 2.33).

Posteriormente es necesario añadir la imagen de la *ramdisk* al *Walrus*.

Al igual que lo indicado en el Código 2.9, el proceso es el mismo con la diferencia de que en este caso (Código 2.10), se está realizando la inserción de la imagen de la *ramdisk* (Figura 2.34, Figura2.35, Figura 2.36).

```
euca-bundle-image -i ~/euca-centos-5.3-i386/kvm-kernel/initrd.img-
```

```

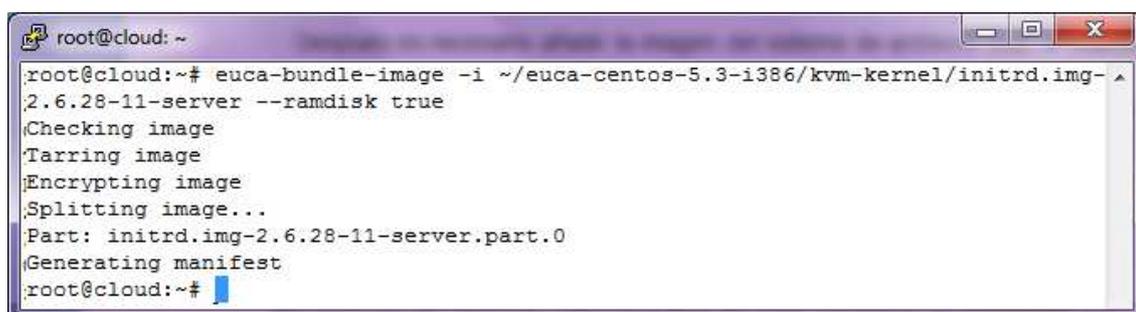
2.6.28-11-server --ramdisk true

euca-upload-bundle -b centos-ramdisk-bucket -m /tmp/initrd.img-2.6.28-
11-server.manifest.xml

euca-register centos-ramdisk-bucket/initrd.img-2.6.28-11-
server.manifest.xml

```

Código 2.10 Imagen Centos: inserción de la imagen (*ramdisk*)

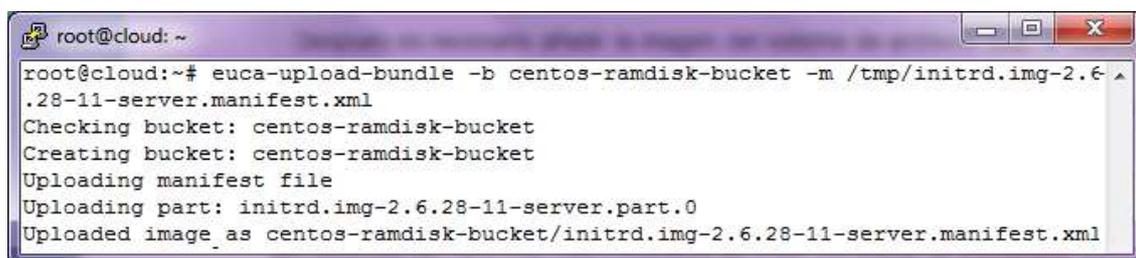


```

root@cloud: ~
root@cloud:~# euca-bundle-image -i ~/euca-centos-5.3-1386/kvm-kernel/initrd.img-
2.6.28-11-server --ramdisk true
Checking image
Tarring image
Encrypting image
Splitting image...
Part: initrd.img-2.6.28-11-server.part.0
Generating manifest
root@cloud:~#

```

Figura 2.34 euca-bundle-image (*ramdisk*)

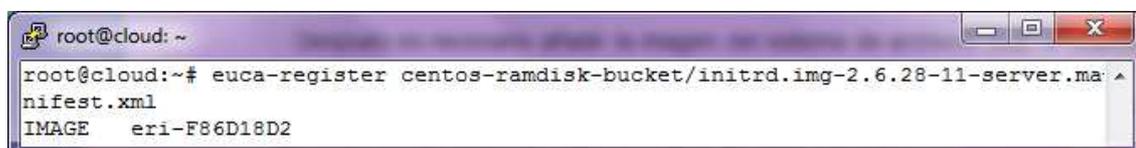


```

root@cloud: ~
root@cloud:~# euca-upload-bundle -b centos-ramdisk-bucket -m /tmp/initrd.img-2.6
28-11-server.manifest.xml
Checking bucket: centos-ramdisk-bucket
Creating bucket: centos-ramdisk-bucket
Uploading manifest file
Uploading part: initrd.img-2.6.28-11-server.part.0
Uploaded image as centos-ramdisk-bucket/initrd.img-2.6.28-11-server.manifest.xml

```

Figura 2.35 euca-upload-bundle (*ramdisk*)



```

root@cloud: ~
root@cloud:~# euca-register centos-ramdisk-bucket/initrd.img-2.6.28-11-server.ma
nifest.xml
IMAGE   eri-F86D18D2

```

Figura 2.36 euca-register (*ramdisk*)

A pesar de que las imágenes certificadas⁶³ de *Eucalyptus*, no requieren de la inserción de la *ramdisk*, se consideró necesario ya que el proceso a seguir es el

⁶³ **Imágenes certificadas** son imágenes creadas y distribuidas por *Eucalyptus Systems*. Son compatibles con *KVM* y con *XEN*.

mismo en el caso de utilizar sistemas operativos que no estén certificados, pero que sean compatibles.

En el Código 2.11, se establece una asociación entre el *kernel* y la *ramdisk* de la imagen creada para que finalmente se genere una imagen de Centos la cual representa a la “plantilla” de Centos para la ejecución de una máquina virtual.

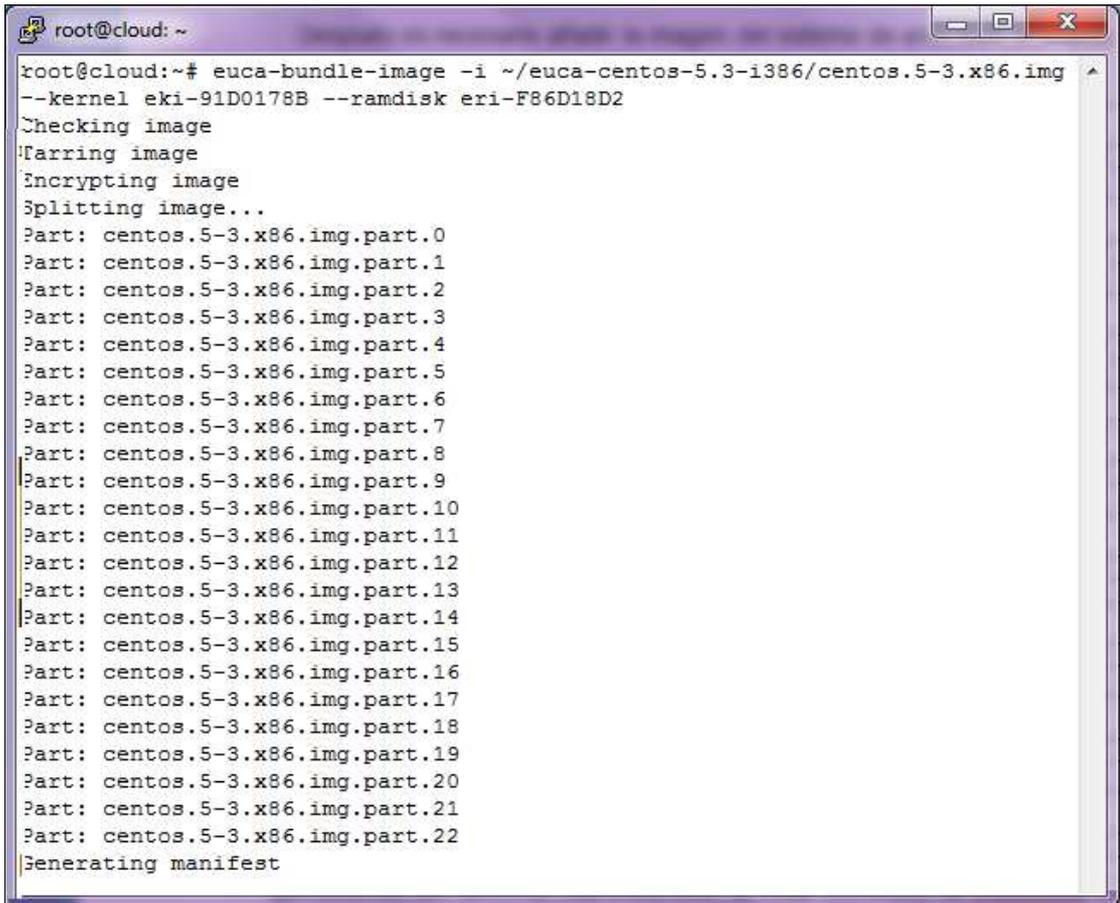
```
euca-bundle-image -i ~/euca-centos-5.3-i386/centos.5-3.x86.img --kernel
eki-91D0178B --ramdisk eri-F86D18D2

euca-upload-bundle -b centos-image-bucket -m /tmp/centos.5-
3.x86.img.manifest.xml

euca-register centos-image-bucket/centos.5-3.x86.img.manifest.xml
```

Código 2.11 Imagen Centos: asociación *kernel/ramdisk*

Se ha dividido a la imagen en varias partes, las cuales junto con el *manifest* generado (Figura 2.37) serán utilizadas por el *Walrus* para el proceso de ejecución de una instancia en los nodos, ya que los archivos de la imagen son copiados en un directorio del nodo.



```
root@cloud: ~  
root@cloud:~# euca-bundle-image -i ~/euca-centos-5.3-i386/centos.5-3.x86.img  
--kernel eki-91D0178B --ramdisk eri-F86D18D2  
Checking image  
Parring image  
Encrypting image  
Splitting image...  
Part: centos.5-3.x86.img.part.0  
Part: centos.5-3.x86.img.part.1  
Part: centos.5-3.x86.img.part.2  
Part: centos.5-3.x86.img.part.3  
Part: centos.5-3.x86.img.part.4  
Part: centos.5-3.x86.img.part.5  
Part: centos.5-3.x86.img.part.6  
Part: centos.5-3.x86.img.part.7  
Part: centos.5-3.x86.img.part.8  
Part: centos.5-3.x86.img.part.9  
Part: centos.5-3.x86.img.part.10  
Part: centos.5-3.x86.img.part.11  
Part: centos.5-3.x86.img.part.12  
Part: centos.5-3.x86.img.part.13  
Part: centos.5-3.x86.img.part.14  
Part: centos.5-3.x86.img.part.15  
Part: centos.5-3.x86.img.part.16  
Part: centos.5-3.x86.img.part.17  
Part: centos.5-3.x86.img.part.18  
Part: centos.5-3.x86.img.part.19  
Part: centos.5-3.x86.img.part.20  
Part: centos.5-3.x86.img.part.21  
Part: centos.5-3.x86.img.part.22  
Generating manifest
```

Figura 2.37 euca-bundle-image asociación (*kernel / ramdisk*)

Se carga el *bucket* resultante de la asociación entre el *kernel* y la *ramdisk* (Figura 2.38) y se lo registra (Figura 2.39) para obtener como resultado el identificador de la imagen de la máquina virtual (emi-XXXXXXX).

```

root@cloud:~# euca-upload-bundle -b centos-image-bucket -m /tmp/centos.5-3.x86
6.img.manifest.xml
Checking bucket: centos-image-bucket
Creating bucket: centos-image-bucket
Uploading manifest file
Uploading part: centos.5-3.x86.img.part.0
Uploading part: centos.5-3.x86.img.part.1
Uploading part: centos.5-3.x86.img.part.2
Uploading part: centos.5-3.x86.img.part.3
Uploading part: centos.5-3.x86.img.part.4
Uploading part: centos.5-3.x86.img.part.5
Uploading part: centos.5-3.x86.img.part.6
Uploading part: centos.5-3.x86.img.part.7
Uploading part: centos.5-3.x86.img.part.8
Uploading part: centos.5-3.x86.img.part.9
Uploading part: centos.5-3.x86.img.part.10
Uploading part: centos.5-3.x86.img.part.11
Uploading part: centos.5-3.x86.img.part.12
Uploading part: centos.5-3.x86.img.part.13
Uploading part: centos.5-3.x86.img.part.14
Uploading part: centos.5-3.x86.img.part.15
Uploading part: centos.5-3.x86.img.part.16
Uploading part: centos.5-3.x86.img.part.17
Uploading part: centos.5-3.x86.img.part.18
Uploading part: centos.5-3.x86.img.part.19
Uploading part: centos.5-3.x86.img.part.20
Uploading part: centos.5-3.x86.img.part.21
Uploading part: centos.5-3.x86.img.part.22
Uploaded image as centos-image-bucket/centos.5-3.x86.img.manifest.xml

```

Figura 2.38 euca-upload-bundle asociación (kernel / ramdisk)

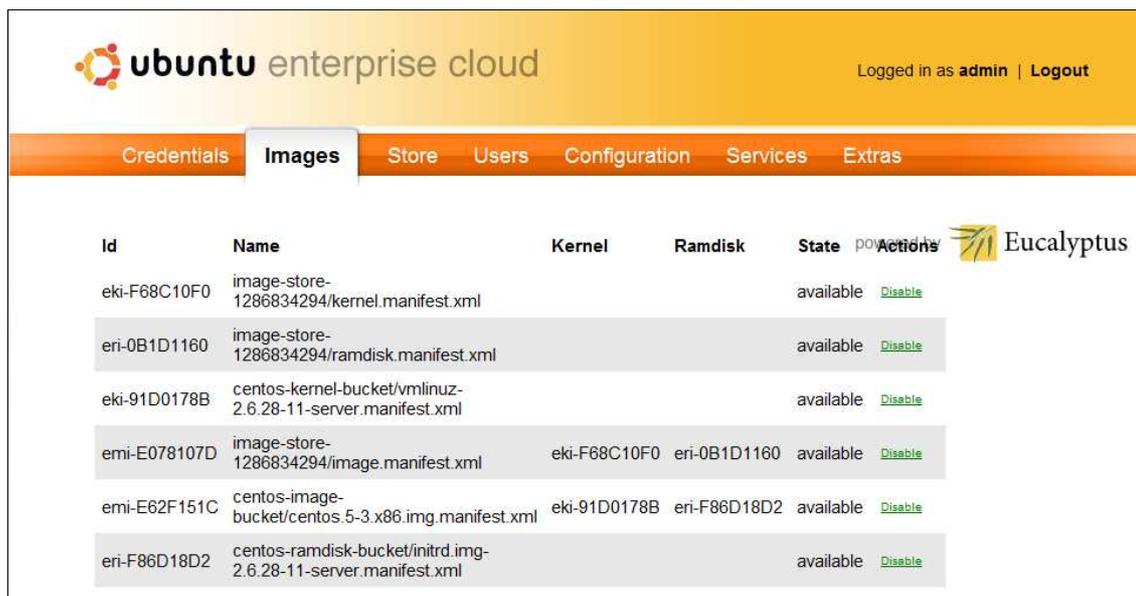
```

root@cloud:~# euca-register centos-image-bucket/centos.5-3.x86.img.manifest.
ml
IMAGE   emi-E62F151C

```

Figura 2.39 euca-register imagen asociación (kernel / ramdisk)

En la interfaz web de UEC (Figura 2.40) se pueden observar las imágenes que ya han sido insertadas y registradas en el sistema, en este caso el identificador emi-E078107D corresponde a la imagen insertada a través del menú “Store” de la interfaz web ya que el nombre corresponde a image-store. El segundo identificador emi-E62F151C (Figura 2.40) corresponde a la imagen de Centos insertada a través de las *euca2ools*, de la misma manera el nombre corresponde a “centos-image”.



Id	Name	Kernel	Ramdisk	State	Actions
eki-F68C10F0	image-store-1286834294/kernel.manifest.xml			available	Disable
eri-0B1D1160	image-store-1286834294/ramdisk.manifest.xml			available	Disable
eki-91D0178B	centos-kernel-bucket/vmlinuz-2.6.28-11-server.manifest.xml			available	Disable
emi-E078107D	image-store-1286834294/image.manifest.xml	eki-F68C10F0	eri-0B1D1160	available	Disable
emi-E62F151C	centos-image-bucket/centos.5-3.x86.img.manifest.xml	eki-91D0178B	eri-F86D18D2	available	Disable
eri-F86D18D2	centos-ramdisk-bucket/initrd.img-2.6.28-11-server.manifest.xml			available	Disable

Figura 2.40 Imágenes disponibles en el sistema

Las imágenes están disponibles para que puedan ser utilizadas para ejecutar máquinas virtuales.

2.5.6 EJECUCIÓN DE INSTANCIAS (MÁQUINAS VIRTUALES)

En esta sección se mostrará como ejecutar una máquina virtual a partir de una imagen insertada en el sistema. Sin embargo, este proceso sólo se realizará para comprobar que todo está funcionando correctamente puesto que del lado del administrador, la tarea de configuración ha sido terminada. En el siguiente capítulo se realizarán las pruebas correspondientes a un usuario común.

Antes de ejecutar una máquina virtual es necesario crear las *keypairs*, las cuales son utilizadas para autenticar usuarios, constan de dos llaves, una es una llave privada, la cual es almacenada en el equipo del usuario, en un archivo que él escoja, y la otra es una llave pública que se almacena en *Eucalyptus*. [53] El Código 2.12 muestra la creación del par de llaves para la cuenta del administrador cuyos nombres se han decidido que sean: *llavesAdmin* (pública) y *llavesAdmin.priv* (privada).

```
mkdir -p -m 700 ~/.euca
touch ~/.euca/llavesAdmin.priv
chmod 0600 ~/.euca/llavesAdmin.priv
euca-add-keypair llavesAdmin > ~/.euca/llavesAdmin.priv
```

Código 2.12 Creación de las llaves *keypairs*

La llave pública le sirve a un usuario para poder ejecutar máquinas virtuales, y la llave privada para acceder a ellas a través de SSH. La llave privada se puede almacenar en cualquier archivo que escoja el usuario, en este caso *llavesAdmin.priv* del directorio (*~/.euca*).

El Código 2.13 muestra que para ejecutar la máquina virtual se utiliza el comando *euca-run-instances* junto con el identificador *emi-E62F151C* correspondiente a la imagen de Centos; la opción *-k* permite especificar la llave pública del usuario, y la opción *-t* especifica de qué tamaño se quiere la máquina virtual (Figura 2.41).

```
cd ~/.euca
euca-run-instances emi-E62F151C -k llavesAdmin -t c1.medium
```

Código 2.13 Ejecución de una máquina virtual

The screenshot shows the 'Configuration' tab of the Ubuntu Enterprise Cloud interface. It displays a table of VM Types with columns for Name, CPUs, Memory (MB), and Disk (GB). The 'c1.medium' type is highlighted. Below the table is a 'Save VmTypes' button.

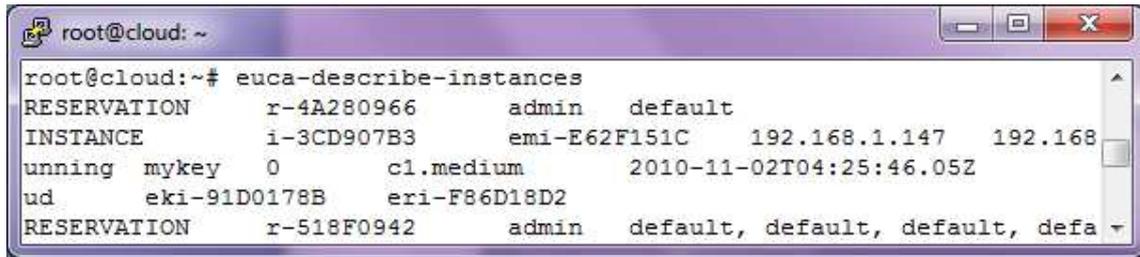
Name	CPUs	Memory (MB)	Disk (GB)
m1.small	1	128	5
c1.medium	1	256	10
m1.large	2	512	15
m1.xlarge	2	1024	20
c1.xlarge	4	2048	20

Save VmTypes

Figura 2.41 Tipos de Máquina Virtual

En el menú “*Configuration*” de la Figura 2.41 se encuentran los tipos (tamaños) de las máquinas virtuales que pueden ser ejecutadas en el sistema, estos pueden ser configurados por el administrador a través de la interfaz web cambiando el valor en las cajas de texto y luego haciendo clic el botón “saveVmTypes”.

En la Figura 2.42 se muestra una máquina virtual en ejecución.



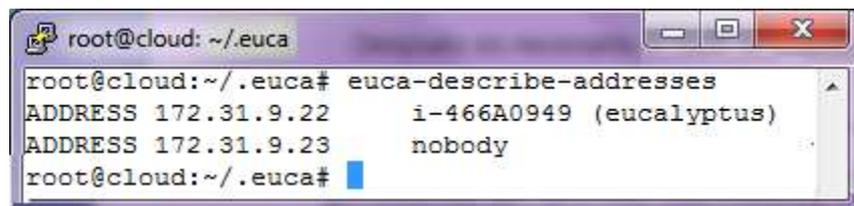
```

root@cloud: ~
root@cloud:~# euca-describe-instances
RESERVATION    r-4A280966      admin  default
INSTANCE       i-3CD907B3      emi-E62F151C  192.168.1.147  192.168
unning mykey  0              c1.medium    2010-11-02T04:25:46.05Z
ud            eki-91D0178B    eri-F86D18D2
RESERVATION    r-518F0942      admin  default, default, default, defa

```

Figura 2.42 Máquina Virtual de Centos ejecutándose

Dado que en el archivo “*eucalyptus.conf*” se configuró un rango de IPs “públicas” (VNET_PUBLICIPS), y (VNET_SUBNET) para el rango de IPs privadas, cada vez que se ejecuta una máquina virtual se le asigna una de las IPs privadas y una de las IPs “públicas” disponibles (Figura 2.43).



```

root@cloud: ~/.euca
root@cloud:~/.euca# euca-describe-addresses
ADDRESS 172.31.9.22      i-466A0949 (eucalyptus)
ADDRESS 172.31.9.23      nobody
root@cloud:~/.euca#

```

Figura 2.43 IPs “públicas” disponibles

En el archivo */var/run/eucalyptus/net/euca-dhcp.conf*, (Figura 2.44), se encuentran las IPs privadas asignadas a las máquinas virtuales a través de DHCP.

MV pasará al estado apagándose (*shutting-down*) y cuando este proceso finalice, la máquina virtual habrá concluido su ciclo de vida. Una vez que la máquina virtual pase al estado terminada (*terminated*), toda la información que se haya producido en ella, se perderá, por lo tanto en el siguiente capítulo se hará uso del componente de *Eucalyptus* llamado *Storage Controller*, el cual permite el almacenamiento persistente de información.



Fuente: [44]

Figura 2.45 Ciclo de vida de una máquina virtual

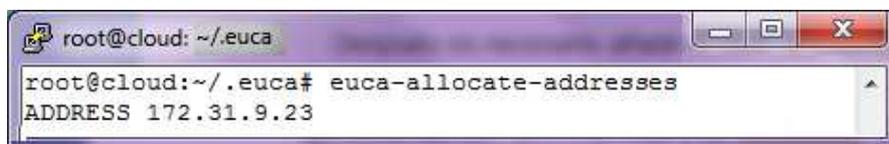
2.5.7 GRUPOS DE SEGURIDAD Y *ELASTIC IP ADDRESSES* EN *EUCALYPTUS*

2.5.7.1 *Elastic IP addresses*

Son direcciones IP estáticas asignadas dinámicamente a partir de un *pool* de direcciones fijas. Una dirección IP elástica puede ser asignada a una cuenta de usuario y éste puede asociarla con una instancia en particular, adicionalmente se tiene el control de esta dirección hasta que se decida liberarla. [74]

En la Figura 2.43 se pueden ver las 2 direcciones IP públicas disponibles, con las que se va a trabajar en el proyecto.

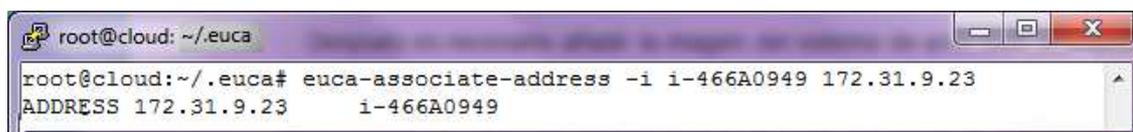
En este ejemplo, en la Figura 2.46 se puede ver la única IP que actualmente está disponible para ser asignada a una máquina virtual puesto que la otra dirección ya está siendo utilizada.



```
root@cloud: ~/.euca
root@cloud:~/.euca# euca-allocate-addresses
ADDRESS 172.31.9.23
```

Figura 2.46 euca-allocate-address

En la Figura 2.47 se muestra la asociación de la dirección IP disponible (Figura 2.46) a una máquina virtual.



```
root@cloud: ~/.euca
root@cloud:~/.euca# euca-associate-address -i i-466A0949 172.31.9.23
ADDRESS 172.31.9.23      i-466A0949
```

Figura 2.47 euca-associate-address

En el caso de que se requiera utilizar esa misma dirección IP, se puede utilizar el comando `euca-disasociate-address` y `euca-release-address`, para desasociar la IP de esa máquina virtual, y para liberar esa IP de la colección de IPs públicas disponibles en el sistema.

2.5.7.2 Grupos de seguridad

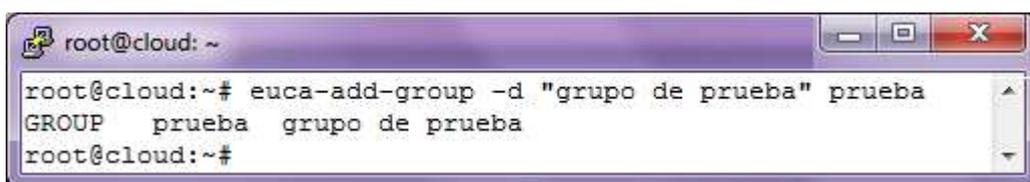
Son aquellos que permiten controlar la seguridad de red, al agrupar a los usuarios de la Nube dentro de distintos grupos que manejen permisos de acceso específicos, ya que para cada grupo se pueden definir protocolos a admitirse (TCP, UDP, ICMP, HTTP, SSH, etc.), número de puertos a utilizar o tipo de servicios que se va a restringir.

Un grupo de seguridad es un conjunto de máquinas virtuales, que permite a un administrador controlar el acceso y uso de dichas máquinas. [53]

Éste es un concepto que se utiliza tanto en la Nube pública de *Amazon*, como en las Nubes privadas desarrolladas con *Eucalyptus*.

Los grupos de seguridad representan una funcionalidad bastante útil a la hora de controlar el tráfico entrante a un grupo de máquinas virtuales pertenecientes a un usuario. El tráfico saliente se puede controlar manualmente con reglas en el *firewall* a través de *iptables*⁶⁴ en (*/var/run/eucalyptus/net/iptables-preload*); mientras que las reglas que se establecen en los grupos de seguridad, se crean automáticamente en el *firewall*.

En la Figura 2.48 se puede observar cómo se crea un grupo. Una vez creado el grupo, se le puede dar permisos de acuerdo a lo que requiera realizar; por ejemplo, se le puede dar autorización para tráfico entrante de ICMP y de TCP proveniente de cualquier dirección IP (Figura 2.49).

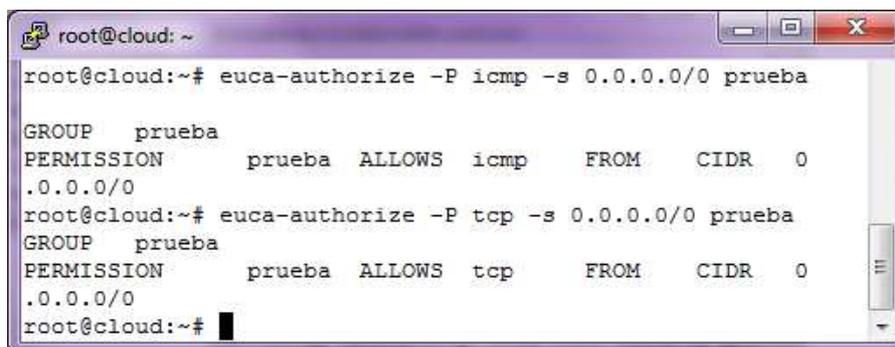


```

root@cloud: ~
root@cloud:~# euca-add-group -d "grupo de prueba" prueba
GROUP   prueba   grupo de prueba
root@cloud:~#

```

Figura 2.48 euca-add-group



```

root@cloud: ~
root@cloud:~# euca-authorize -P icmp -s 0.0.0.0/0 prueba
GROUP   prueba
PERMISSION   prueba  ALLOWS  icmp   FROM   CIDR   0
.0.0.0/0
root@cloud:~# euca-authorize -P tcp -s 0.0.0.0/0 prueba
GROUP   prueba
PERMISSION   prueba  ALLOWS  tcp    FROM   CIDR   0
.0.0.0/0
root@cloud:~# █

```

Figura 2.49 euca-authorize

En la Figura 2.49, la opción `-P` especifica el protocolo y la opción `-s` especifica la IP de origen, y finalmente se especifica el grupo al cual se quiere aplicar la regla.

Cada máquina virtual puede utilizar su propio *firewall* para controlar el tráfico estableciendo reglas (*iptables*). Sin embargo, el uso de los grupos de seguridad

⁶⁴ *iptables* son el sistema de *firewall* vinculado al *kernel* de Linux.

es más conveniente ya que en cada grupo se ha establecido un conjunto de permisos que pueden ser aplicados a una colección de máquinas virtuales que se ejecutan en dicho grupo.

El comando `euca-revoke` permite quitar la regla al grupo, la sintaxis es la misma; de igual forma, el comando `euca-delete-group` permite borrar un grupo.

Cuando se requiere ejecutar una máquina virtual con un determinado grupo, se utiliza la opción `-g` en `euca-run-instances`.

2.5.8 INSTALACIÓN Y CONFIGURACIÓN DE *ELASTICFOX*

Elasticfox es una herramienta que permite gestionar los recursos de una Nube privada de *Eucalyptus* y de la Nube pública de *Amazon*, ya que brinda acceso a las cuentas de *Eucalyptus* y *Amazon Web Services (AWS)*. [77]

Es un *plug-in*⁶⁵ para el navegador *Mozilla Firefox*, y permite a los administradores de la Nube configurar la capacidad de la misma bajo la comodidad de una interfaz de navegador web. [77]

Instalación, inicialización y configuración de *Elasticfox*

1. Instalar Mozilla Firefox en Windows o Linux, desde www.mozilla.com.
2. Una vez instalado el navegador *Mozilla Firefox*, se debe instalar la versión actual de *Elasticfox*, como se muestra en la Figura 2.50, se debe navegar a la página oficial de *Amazon Web Services*: [79]

<http://developer.amazonwebservices.com/connect/entry.jspa?externalID=609&categoryID=88>

⁶⁵ *plug-in* es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. [78]

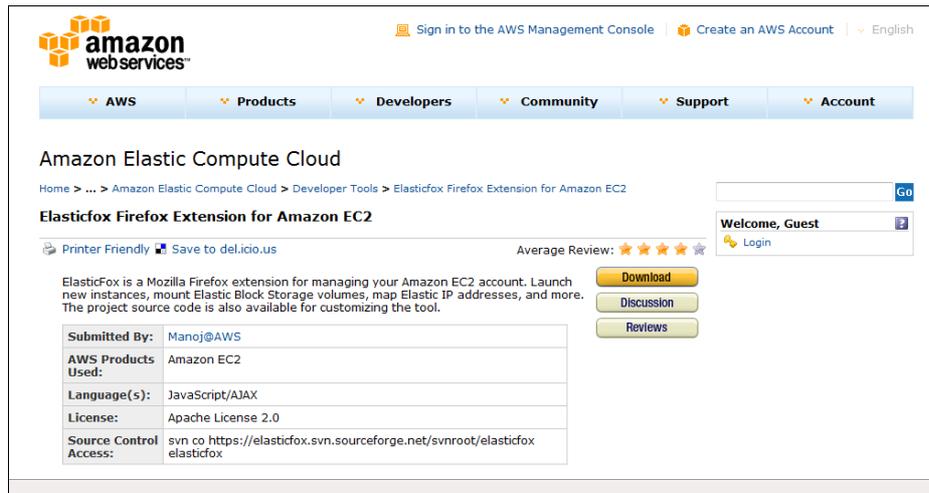


Figura 2.50 Amazon Web Services, *Elasticfox*

Descargar e instalar el archivo ***elasticfox.xpi*** (Figura2.52)

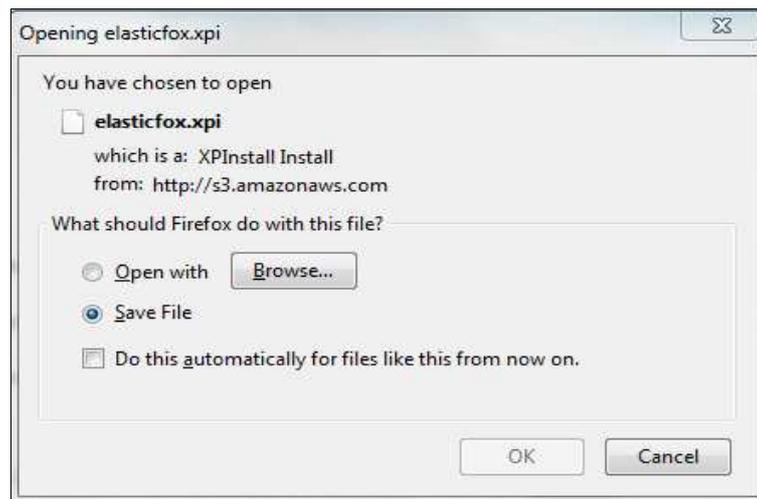


Figura 2.51 Archivo *elasticfox.xpi*

Una vez que el *plug-in* ha sido instalado, en el menú "Tools", aparecerá *Elasticfox* (Figura 2.52) con lo que ya se puede acceder a esta herramienta de gestión.

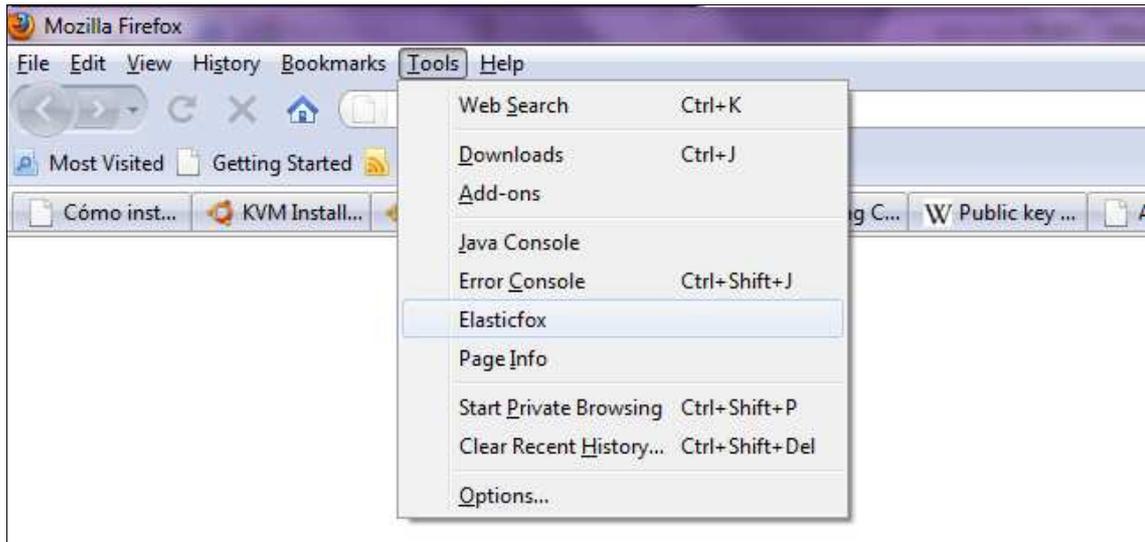


Figura 2.52 Elasticfox

3. Configuración de *Elasticfox*

En la Figura 2.53 se deben configurar las credenciales obtenidas de la cuenta de un usuario accediendo a través de la URL: <https://cloud.epn.edu.ec:8443> (Figura 2.27, "Query interface credentials").

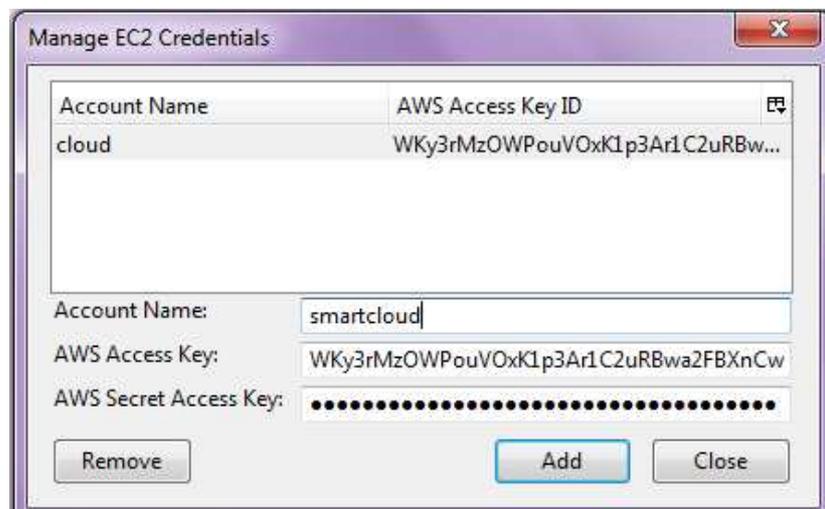


Figura 2.53 Configuración de credenciales en *Elasticfox*

Luego de configurar las credenciales de un usuario de la Nube, se deben configurar las regiones (Figura 2.54) en las cuales se van a ejecutar las máquinas virtuales; la dirección a la cual se va a conectar *Elasticfox* es

<http://cloud.epn.edu.ec:8773/services/Eucalyptus>, establecida en la variable EC2_URL del archivo eucarc (Código 2.6).

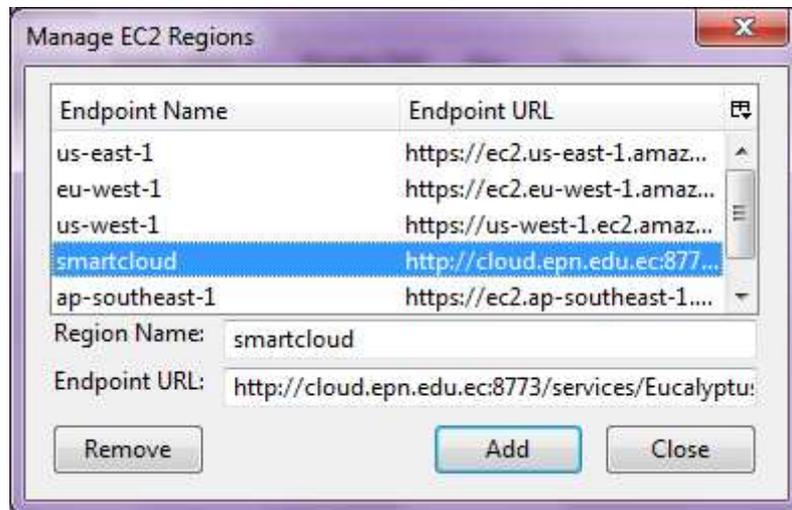


Figura 2.54 Configuración de *Regions* en *Elasticfox*

Para establecer finalmente la conexión con la Nube (Figura 2.55), se debe establecer la variable `ec2ui.url` de *Elasticfox* en: <http://cloud.epn.edu.ec:8773/services/Eucalyptus>, con un valor igual al de la variable EC2_URL del archivo eucarc, para que al iniciar *Elasticfox* se conecte al CLC de la Nube. Esto se realiza accediendo a las variables del navegador web con (about:config).

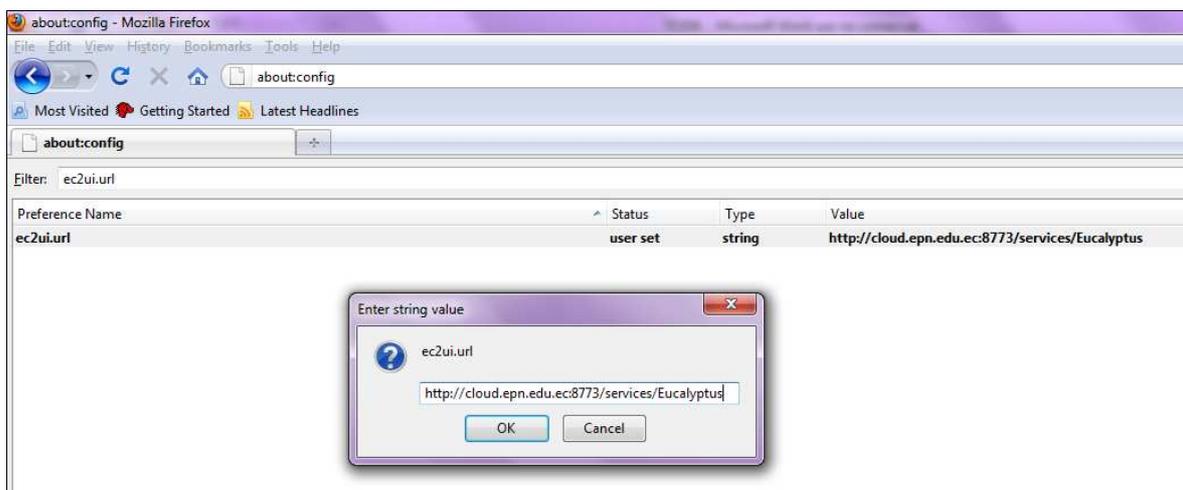


Figura 2.55 Configuración de *Regions* en *Elasticfox*

Si se tiene acceso a la administración de la Nube, dependiendo del tipo de permisos que posea cada usuario; en la Figura 2.56 se pueden observar las imágenes con las cuales puede trabajar el usuario Administrador ya que las credenciales que se añadieron a *Elasticfox* corresponden a este usuario; si se desea trabajar con otro usuario, se deberán añadir las nuevas credenciales. En el siguiente capítulo se utilizará *Elasticfox* con un usuario que no tenga privilegios de Administrador.

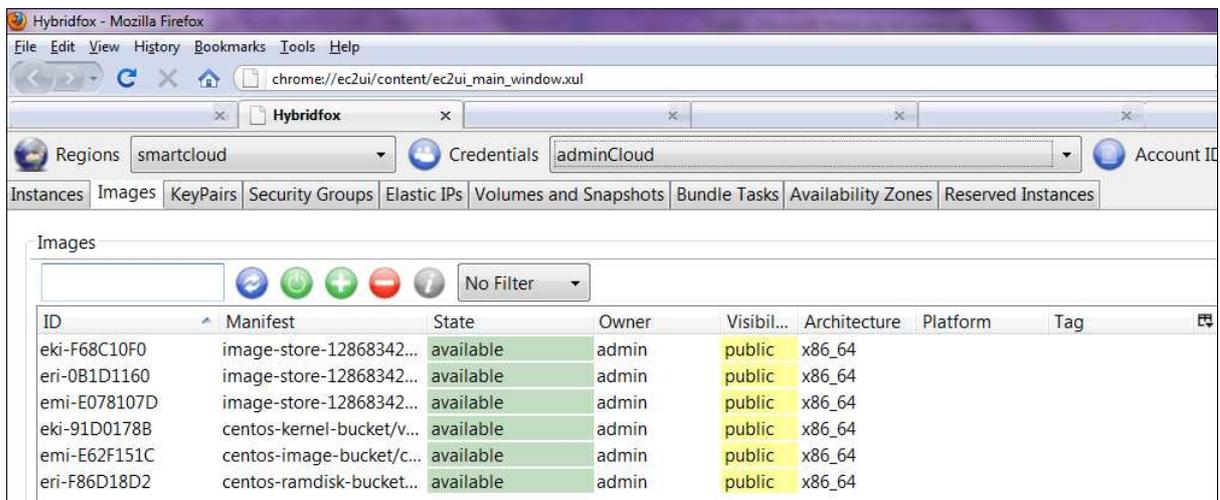


Figura 2.56 Imágenes vistas desde *Elasticfox*

Se debe tomar en cuenta, que desde el navegador web, utilizando *Elasticfox*, se podrán ver y manejar todos los elementos de la Nube, incluyendo instancias creadas, imágenes disponibles en el sistema, configuraciones de seguridad, direcciones IP disponibles, etc., de la misma manera que se administraría la Nube a través de un cliente SSH.

2.6 COSTOS REFERENCIALES DE LOS ELEMENTOS UTILIZADOS PARA LA REALIZACIÓN DEL PRESENTE PROYECTO

Como se describió en la Sección 2.5.2, el prototipo de *Cloud Computing* consta de un *front-end* y dos nodos. Adicionalmente se utilizan equipos para la interconexión de los dispositivos de red.

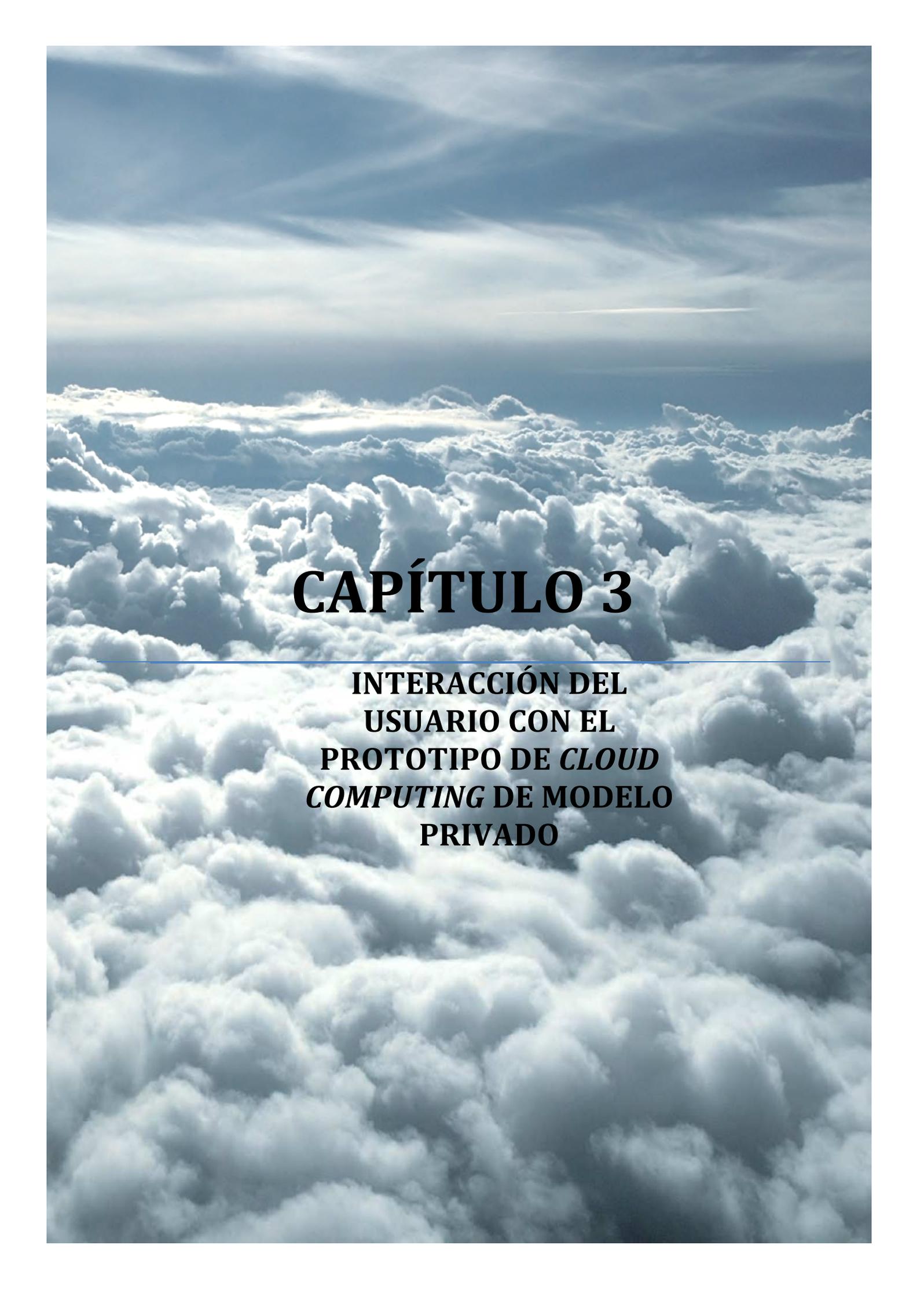
En la Tabla 2.4 se especifica el valor de cada uno de los elementos utilizados en la implementación del prototipo. Se han tomado como referencia los valores que se encuentran en el Anexo F.

Nombre Equipo	Descripción	Cantidad	V. Unitario	V. Total (USD ⁶⁶)
Computador Front-end	- Procesador Dual Core - Memoria RAM 4GB - Disco Duro Sata - Monitor 17"	1	498.00	498.00
Computadores Nodos	- INTEL-CLONE ALTEK - Procesador Intel Core 2 Quad 2.6 GHz 4MB - Memoria RAM DDR2 800 MHz 8 GB (4x2 Gb) - 2 Discos Duros 500 GB Sata - Regulador de voltaje	2	665.00	1330.00
Switch	- CNET - CGS-800 - Gigabit Switch - 8 puertos	1	71.50	71.50
Cable UTP	- CAT 5	10 m	0.60	6.00
Switch KVM	- KVM Switch DLINK - 4 puertos - DKVM – 4K	1	69.99	69.99
COSTO TOTAL				\$1975.49

Tabla 2.4 Costos de los elementos utilizados para la implementación de este prototipo

Por lo tanto, el costo total de los elementos utilizados para la implementación del prototipo es de \$1975.49.

⁶⁶ Dólares americanos.



CAPÍTULO 3

**INTERACCIÓN DEL
USUARIO CON EL
PROTOTIPO DE *CLOUD*
COMPUTING DE MODELO
PRIVADO**

3.1 CUENTAS DE USUARIO

Los usuarios, imágenes, *clusters*, grupos, información específica del *Walrus* como: *buckets*, objetos, permisos, información específica del *Storage Controller* (volúmenes, *snap-shots*, etc.), casi toda la información de estado del *front-end* se encuentra almacenada en una base de datos que es del tipo HSQLDB⁶⁷. Todos los *scripts* de la base de datos que son ejecutados para registrar la información en *Eucalyptus* se pueden encontrar en el directorio `/var/lib/eucalyptus/db`. [82]

Un usuario puede acceder a una cuenta a partir de la interfaz de Inicio de sesión del sistema, <https://cloud.epn.edu.ec:8443>, para lo cual existe el enlace que permite aplicar (“*Apply*”, Figura 3.1) a una nueva cuenta; este abrirá una ventana donde se mostrará un formulario (Figura 3.2) que debe ser llenado por el nuevo usuario, para que se envíe a la cuenta de correo electrónico del administrador y así éste sea aprobado para ingresar al sistema con sus respectivas credenciales.



ubuntu enterprise cloud
Version 1.6
Please, sign in:
Username:
Password:
 Remember me on this computer
Sign in
[Apply](#) for account | [Recover](#) the Password

Figura 3.1 Ventana de autenticación de usuarios

⁶⁷ **HSQLDB** (*Hyperthreaded Structured Query Language Database*) es un sistema gestor de bases de datos libre escrito en Java.



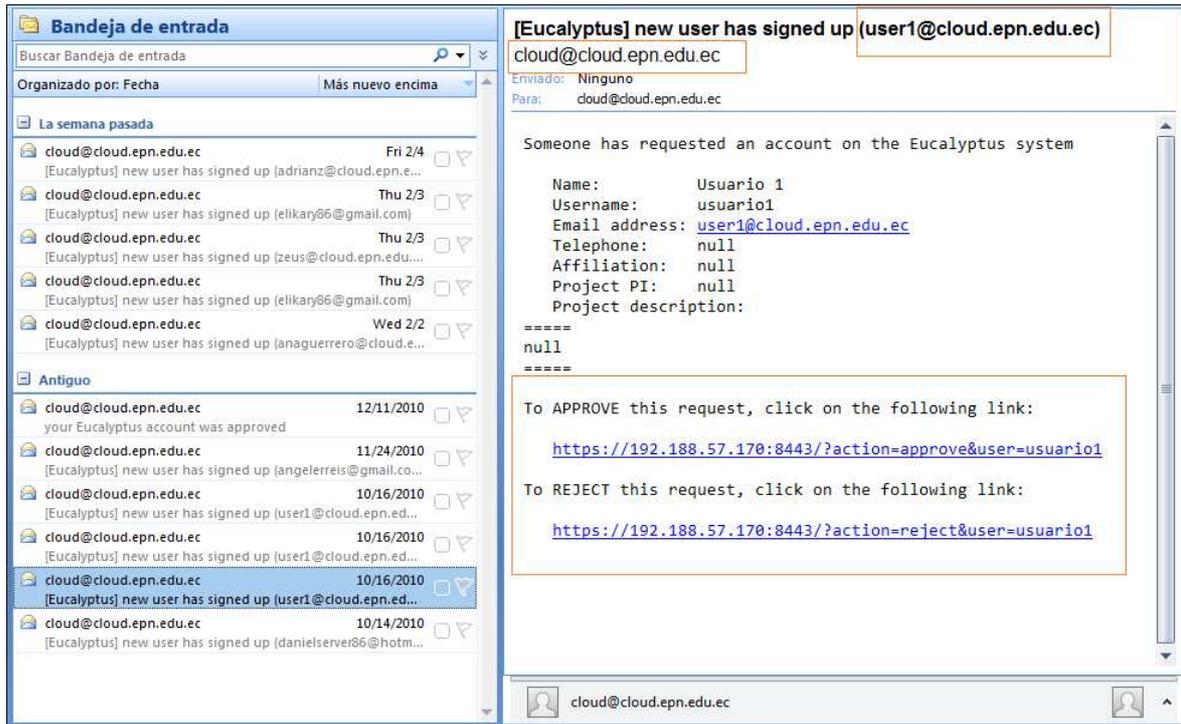
The image shows a registration form for Ubuntu Enterprise Cloud. At the top, there is the Ubuntu logo and the text "ubuntu enterprise cloud". Below this, it says "Please, fill out the form:". The form is divided into two sections: "Mandatory fields:" and "Optional fields:". The mandatory fields include Username, Password, Password, again, Full Name, and Email address. The optional fields include Telephone Number, Project Leader, Affiliation, and Project Description. At the bottom of the form, there are two buttons: "Sign up" and "Cancel", separated by the word "OR".

Figura 3.2 Formulario de aplicación para una nueva cuenta

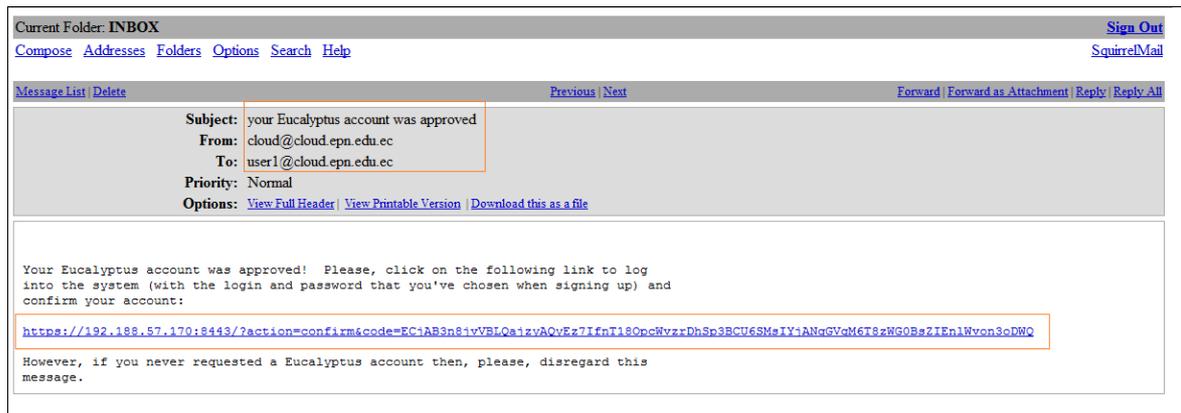
El proyecto implementado utiliza correos electrónicos dentro del dominio cloud.epn.edu.ec, así que es necesario tener una cuenta para poder enviar y recibir correos.

El usuario "usuario1" tiene la cuenta de correo electrónico: user1@cloud.epn.edu.ec, de tal forma que cuando requiera una cuenta en el sistema de Nube, todos los correos de notificación serán enviados a esa cuenta. En la Figura 3.3, se observa el correo electrónico de notificación de que la cuenta ha sido aprobada; el remitente es cloud@cloud.epn.edu.ec, que corresponde a la cuenta del Administrador.

Se puede acceder a la cuenta de correo del Administrador a través de configuraciones en Outlook 2007 (Figura 3.3a, cloud.epn.edu.ec servidor de correo entrante y saliente y POP3) o a través de Squirrelmail (<http://cloud.epn.edu.ec/squirrelmail>). En este caso la Figura 3.3b corresponde a la cuenta del user1.



(a)



(b)

Figura 3.3 Aprobación de una cuenta de usuario a través de correo electrónico

El sistema envía correos de notificación utilizando la cuenta del administrador (cloud@cloud.epn.edu.ec) al mismo administrador (Figura 3.3 a) informando que alguien ha realizado una petición de cuenta, en este caso el “usuario 1” es quien realizó la petición. En el mismo correo de notificación también se envían dos enlaces para aprobar o rechazar la cuenta pedida, así que queda en manos del administrador la aprobación o rechazo de un usuario; en el caso de que se apruebe se envía al nuevo usuario “usuario 1” un correo de notificación de que la

cuenta ha sido aprobada (Figura 3.3 b), cuando el usuario da clic en el enlace enviado, automáticamente se activa su cuenta pudiendo así, tener acceso al sistema.

El usuario puede ingresar al sistema con su cuenta a través de la URL: <https://cloud.epn.edu.ec:8443>, para lo cual deberá ingresar su nombre de usuario y contraseña (Figura 3.4).

The screenshot displays the 'ubuntu enterprise cloud' interface. At the top right, it indicates the user is 'Logged in as user1 | Logout'. A navigation bar contains 'Credentials' and 'Images'. The main content area is titled 'User account Information' and includes the following details:

- Login: **user1**
- Name: **Usuario1**
- Email: **user1@cloud.epn.edu.ec**

Below this information, there is a note: 'Feel free to change the account information (except the login) and the password whenever you want. The cryptographic credentials for the Web services associated with this account, shown below, will not be affected by these changes.' Two buttons are provided: 'Edit Account Information' and 'Change Password'.

The next section is 'Credentials ZIP-file', which states: 'Click the button to download a ZIP file with your Eucalyptus credentials. Use the public/private key pair included therein with tools that require X.509 certificates, such as Amazon's EC2 command-line tools.' A 'Download Credentials' button is present.

The final section is 'Query interface credentials', which says: 'Use this pair of strings with tools - such as [euca2ools](#) - that utilize the "query interface" in which requests and parameters are encoded in the URL.' It lists 'Query ID:' and 'Secret Key:' with a 'Show keys' button.

Figura 3.4 Ingreso a la cuenta del usuario 1 en UEC

En el menú “*Images*” (Figura 3.4), el usuario puede encontrar las imágenes disponibles para él en el sistema, con las cuales podrá ejecutar máquinas virtuales. También puede descargar sus credenciales para tener acceso a la Nube con una herramienta de gestión como *Elasticfox*.

The screenshot shows the Ubuntu Enterprise Cloud administration interface. At the top, there is a navigation bar with the Ubuntu logo and the text 'ubuntu enterprise cloud'. On the right side of the navigation bar, it says 'Logged in as admin | Logout'. Below the navigation bar, there are several tabs: 'Credentials', 'Images', 'Store', 'Users', 'Configuration', 'Services', and 'Extras'. The 'Users' tab is currently selected. Below the tabs, there is a section titled 'Users' with the text 'powered by Eucalyptus'. Below this, there is a table with the following columns: 'Username', 'Email', 'Name', 'Status', and 'Actions'. The table contains two rows of data:

	Username	Email	Name	Status	Actions
3	user1	user1@cloud.epn.edu.ec	Usuario1	active	Edit Disable Delete
2	admin	cloud@cloud.epn.edu.ec	Administrador	active & admin	Edit

Figura 3.5 Cuentas de usuario activas

En la Figura 3.5 se pueden ver las cuentas de usuario activas en el sistema. Este acceso es desde la cuenta del administrador, de esta manera se muestra como el administrador tiene control total sobre todas las cuentas.

3.2 UTILIZACIÓN DE HERRAMIENTAS DE GESTIÓN A NIVEL DE USUARIO

3.2.1 INSTALACIÓN DE *EUCA2TOOLS*

Como se mencionó en el capítulo anterior, estas herramientas de línea de comandos, le permitirán al usuario acceder a los servicios que el *Cloud Controller* ofrece, esta vez con los permisos de un usuario normal al contrario de lo que se realizó anteriormente cuando se utilizó dichas herramientas con los permisos de un administrador (sección 2.5.4.1).

A continuación (Figura 3.6) se muestran los resultados de la instalación de las *euca2ools* en un sistema operativo Ubuntu Desktop 10.04.

```

root@elikary-desktop: ~
File Edit View Terminal Help

root@elikary-desktop:~# apt-get install euca2ools
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  cloud-utils
The following NEW packages will be installed:
  cloud-utils euca2ools
0 upgraded, 2 newly installed, 0 to remove and 330 not upgraded.
Need to get 0B/92.0kB of archives.
After this operation, 655kB of additional disk space will be used.
Do you want to continue [Y/n]? y
Selecting previously deselected package euca2ools.
(Reading database ... 125767 files and directories currently installed.)
Unpacking euca2ools (from .../euca2ools_1.2-0ubuntu10_all.deb) ...
Selecting previously deselected package cloud-utils.
Unpacking cloud-utils (from .../cloud-utils_0.11-0ubuntu1_all.deb) ...
Processing triggers for man-db ...
Setting up euca2ools (1.2-0ubuntu10) ...

Processing triggers for python-central ...
Setting up cloud-utils (0.11-0ubuntu1) ...

```

Figura 3.6 Instalación *euca2ools* nivel usuario

Después es necesario descargar las credenciales al igual que en la sección 2.5.4; este proceso se lo realizó desde la interfaz web accediendo con la cuenta del usuario 1 (Figura 3.4) dando clic en el botón “*Download Credentials*”.

```

root@elikary-desktop: ~/Downloads/euca2-user1-x509
File Edit View Terminal Help

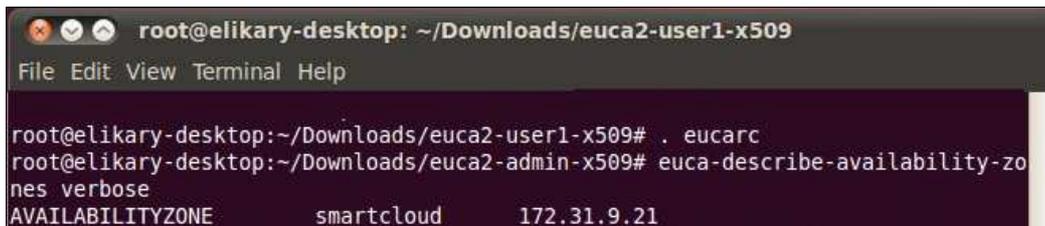
root@elikary-desktop:~/Downloads/euca2-user1-x509# ls
cloud-cert.pem          euca2-user1-f427d54c-pk.pem
euca2-user1-f427d54c-cert.pem  eucarc
root@elikary-desktop:~/Downloads/euca2-user1-x509#
EUCA_KEY_DIR=$(dirname $(readlink -f ${BASH_SOURCE}))
export S3_URL=http://172.31.9.21:8773/services/Walrus
export EC2_URL=http://172.31.9.21:8773/services/Eucalyptus
export EC2_PRIVATE_KEY=${EUCA_KEY_DIR}/euca2-user1-f427d54c-pk.pem
export EC2_CERT=${EUCA_KEY_DIR}/euca2-user1-f427d54c-cert.pem
export EUCALYPTUS_CERT=${EUCA_KEY_DIR}/cloud-cert.pem
export EC2_ACCESS_KEY='N7mvtGjoFW137S7sflK0FYeg3TTUPKHCmScWvw'
export EC2_SECRET_KEY='CA22R1b6nEfQ3FST20qmlJDK00y4bIGLTQZadQ'
# This is a bogus value; Eucalyptus does not need this but client tools do.
export EC2_USER_ID='000106824177'
alias ec2-bundle-image="ec2-bundle-image --cert ${EC2_CERT} --privatekey ${EC2_PRIVATE_KEY} --user 000106824177 --ec2cert ${EUCALYPTUS_CERT}"
alias ec2-upload-bundle="ec2-upload-bundle -a ${EC2_ACCESS_KEY} -s ${EC2_SECRET_KEY} --url ${S3_URL} --ec2cert ${EUCALYPTUS_CERT}"
~
~
~
"eucarc" 12L, 859C                               1,1      All

```

Figura 3.7 Archivo *eucarc* usuario1

En la figura 3.7 se pueden ver todos los certificados (cloud-cert.pem, euca2user1-f427d54c-pk.pem, euca2user1-f427d54c-cert.pem) del usuario 1 que serán utilizados por el script *eucaarc* para establecer las variables de entorno que se utilizan con las *euca2ools*; el archivo *eucaarc* se genera automáticamente cuando se activa la cuenta de un usuario.

En la Figura 3.8 se muestran las *availability zones* de *smartcloud* vistas por el usuario1 en base a sus credenciales especificadas en el archivo *eucaarc*. El comando *euca-describe-availability-zones verbose* es parte de las *euca2ools* y permite mostrar el *cluster* al cual el usuario 1 tiene acceso para ejecutar máquinas virtuales. Para usar las *euca2ools* es necesario ejecutar el script (*source eucaarc*).



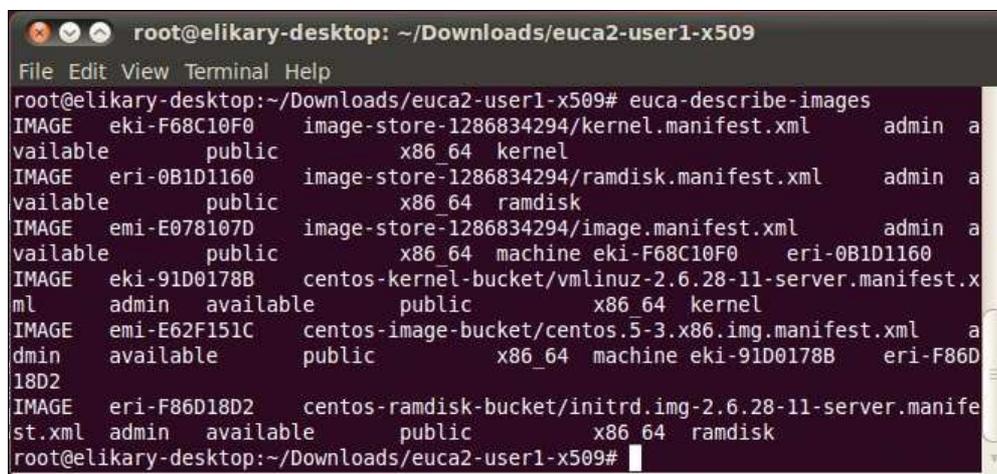
```

root@elikary-desktop: ~/Downloads/euca2-user1-x509
File Edit View Terminal Help
root@elikary-desktop:~/Downloads/euca2-user1-x509# . eucaarc
root@elikary-desktop:~/Downloads/euca2-admin-x509# euca-describe-availability-zones verbose
AVAILABILITYZONE      smartcloud      172.31.9.21

```

Figura 3.8 Availability zones con credenciales de usuario 1

Al igual que el Administrador, los usuarios tienen acceso a las imágenes con las cuales podrán ejecutar instancias ya sea utilizando herramientas de línea de comandos (Figura 3.9), o utilizando herramientas de interfaz gráfica como *Elasticfox*.



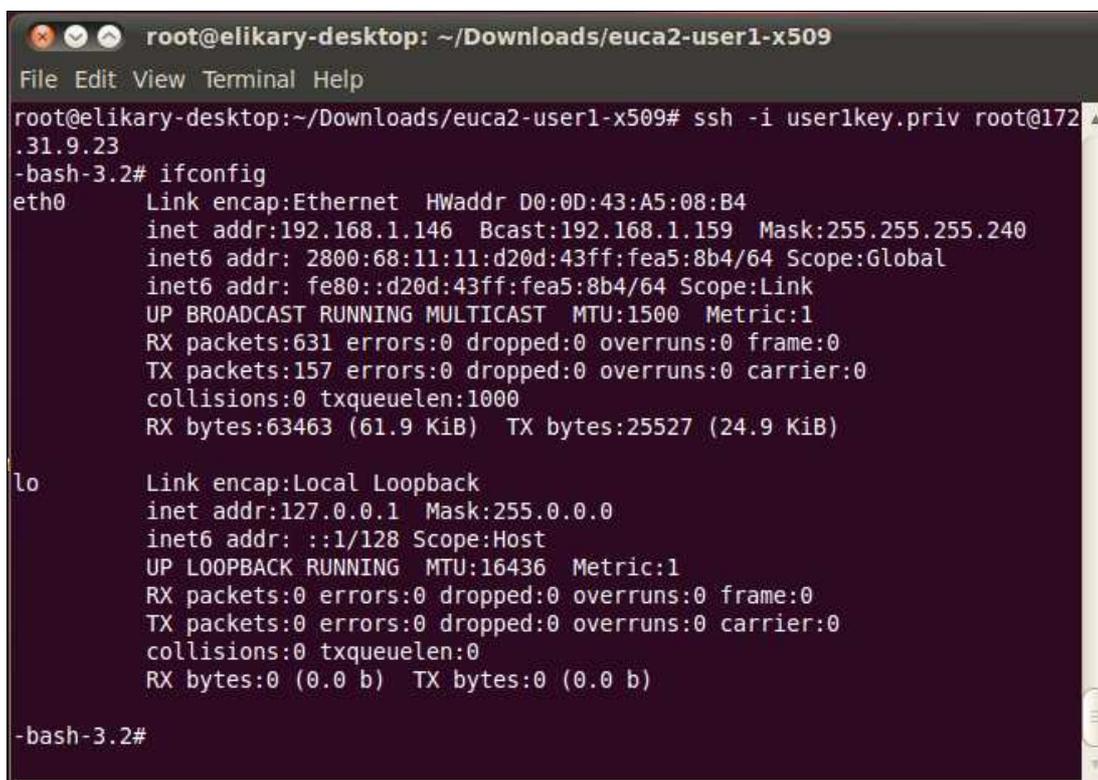
```

root@elikary-desktop: ~/Downloads/euca2-user1-x509
File Edit View Terminal Help
root@elikary-desktop:~/Downloads/euca2-user1-x509# euca-describe-images
IMAGE      eki-F68C10F0      image-store-1286834294/kernel.manifest.xml      admin      a
vailable      public      x86_64      kernel
IMAGE      eri-0B1D1160      image-store-1286834294/ramdisk.manifest.xml      admin      a
vailable      public      x86_64      ramdisk
IMAGE      emi-E078107D      image-store-1286834294/image.manifest.xml      admin      a
vailable      public      x86_64      machine      eki-F68C10F0      eri-0B1D1160
IMAGE      eki-91D0178B      centos-kernel-bucket/vmlinuz-2.6.28-11-server.manifest.xml      admin      available      public      x86_64      kernel
IMAGE      emi-E62F151C      centos-image-bucket/centos.5-3.x86.img.manifest.xml      admin      available      public      x86_64      machine      eki-91D0178B      eri-F86D18D2
IMAGE      eri-F86D18D2      centos-ramdisk-bucket/initrd.img-2.6.28-11-server.manifest.xml      admin      available      public      x86_64      ramdisk
root@elikary-desktop:~/Downloads/euca2-user1-x509#

```

Figura 3.9 Imágenes disponibles, vistas con credenciales del usuario1

En el caso de que utilice la línea de comandos, el cliente SSH permitirá el acceso a una instancia en ejecución (Figura 3.10), utilizando la llave privada del usuario `user1key.priv`, para lo cual es necesario establecer las variables de entorno ejecutando el *script* “`eucarc`” del usuario 1 (Figura 3.7).



```

root@elikary-desktop: ~/Downloads/euca2-user1-x509
File Edit View Terminal Help
root@elikary-desktop:~/Downloads/euca2-user1-x509# ssh -i user1key.priv root@172
.31.9.23
-bash-3.2# ifconfig
eth0      Link encap:Ethernet  HWaddr D0:0D:43:A5:08:B4
          inet addr:192.168.1.146  Bcast:192.168.1.159  Mask:255.255.255.240
          inet6 addr: 2800:68:11:11:d20d:43ff:fea5:8b4/64  Scope:Global
          inet6 addr: fe80::d20d:43ff:fea5:8b4/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:631 errors:0 dropped:0 overruns:0 frame:0
          TX packets:157 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:63463 (61.9 KiB)  TX bytes:25527 (24.9 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)

-bash-3.2#

```

Figura 3.10 Acceso a una instancia en ejecución con llave privada de usuario1 utilizando *euca2ools*

3.2.2 UTILIZACIÓN DE *ELASTICFOX* PARA CONEXIÓN DE SSH CON *PUTTY*⁶⁸ Y CREACIÓN DE LLAVES (*KEYPAIRS*)

La flexibilidad de *Eucalyptus* en cuanto al uso de herramientas administrativas, es bastante amplia, de tal forma que se puede acceder a sus servicios ya sea desde un sistema Linux o Windows, desde línea de comandos o desde una interfaz gráfica.

⁶⁸ *Putty* es un cliente SSH, Telnet, rlogin, y TCP raw con licencia libre. Disponible para Windows y algunas plataformas Unix.



Figura 3.11 Creación de *keypairs* desde *Elasticfox*

Al igual que en la Sección 2.5.6; en la Figura 3.11 se utiliza *Elasticfox* para la creación de las *keypairs*, la llave pública se almacena en *Eucalyptus* y la llave privada se almacena en algún directorio que el usuario escoja, en este caso el archivo que contiene la llave privada del usuario tiene la extensión (pem), el cual será útil para establecer un inicio de sesión desde *Putty*, en una máquina virtual ejecutada en la Nube.

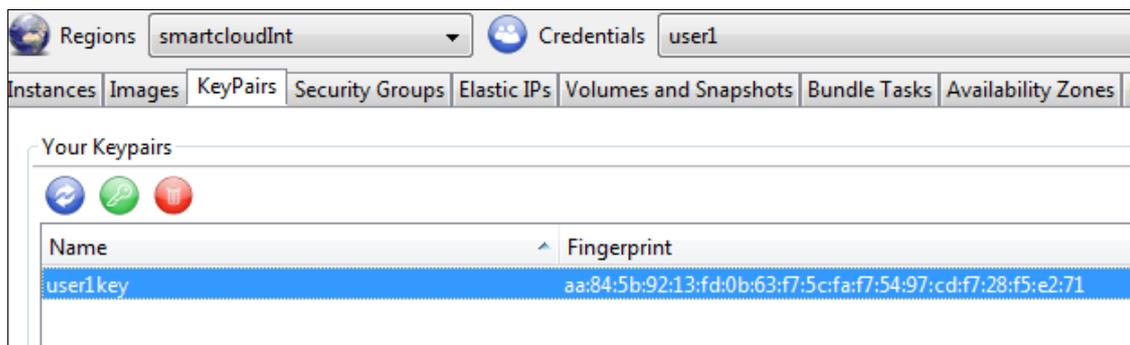


Figura 3.12 *Keypairs* del usuario1 vistas desde *Elasticfox*

La Figura 3.12 presenta las *keypairs* del usuario1, creadas desde *Elasticfox*.

En la Figura 3.13 se puede observar una instancia ejecutándose, tomado en cuenta que las credenciales que se utilizan para esto son del usuario 1. Esta operación se puede realizar utilizando las *euca2ools* o como en el ejemplo, *Elasticfox*.

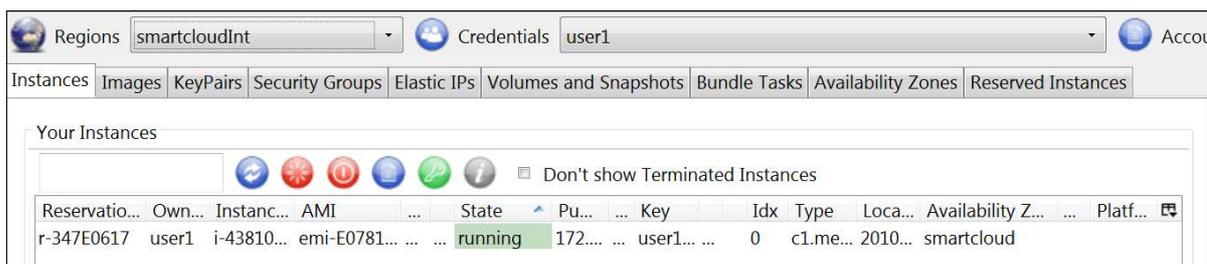


Figura 3.13 Instancia ejecutándose con credenciales del usuario1

Se va a utilizar el cliente SSH *Putty*, desde Windows 7, para acceder remotamente a la instancia creada, para lo cual es necesario contar con la llave privada que forma parte de las *keypairs*.

Putty requiere llaves privadas que tengan la extensión (ppk) de tal forma que la llave almacenada con extensión (pem, de *Elasticfox*) y la llave creada con las *euca2ools* almacenada con extensión (priv) deberán ser convertidas a la extensión (ppk), para lo cual se puede utilizar el *Puttygen*⁶⁹.

Cuando las llaves privadas están listas, es importante configurar los parámetros que utilizará el cliente SSH para conectarse a la máquina virtual. En el botón *Tools* de *Elasticfox* se despliega una ventana (Figura 3.14) con todas las variables necesarias para la conexión.

Ya que *Elasticfox* también se puede conectar con otros sistemas de Nube como Amazon (AWS), consta de parámetros que en este proyecto no han sido utilizados, como por ejemplo RDP Command y RDP Arguments que son útiles para establecer conexiones con máquinas virtuales Windows.

⁶⁹ *Puttygen* es una aplicación que permite generar llaves RSA Y DSA que pueden ser utilizadas con *Putty*.

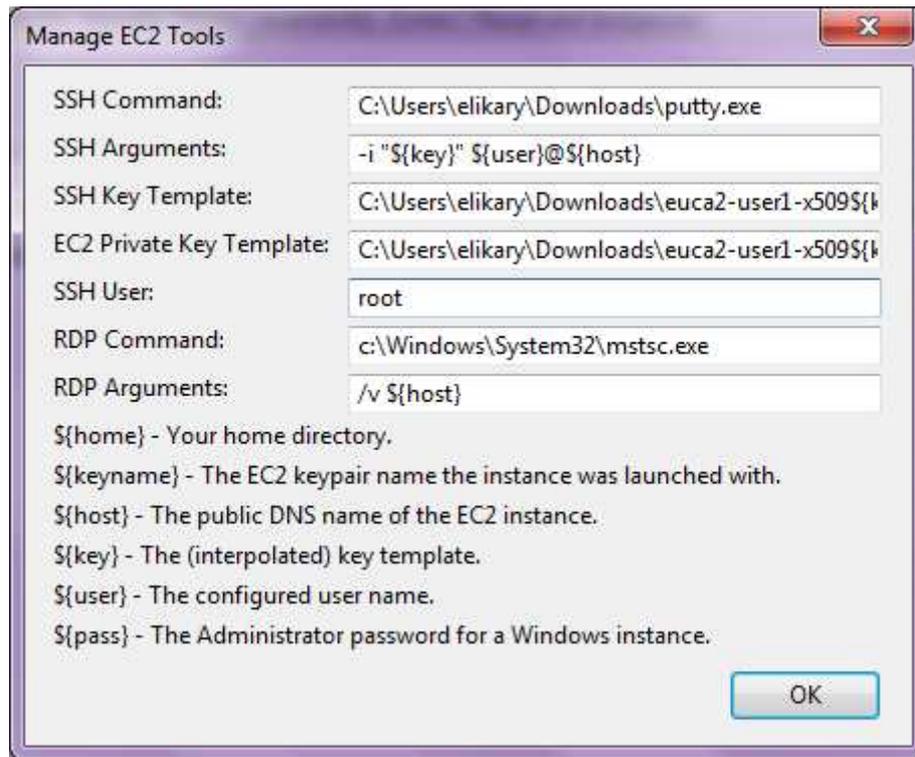


Figura 3.14 Configuración para *Putty* en *Elasticfox*

Entre las variables más importantes están: *SSH Command*, la ubicación del archivo ejecutable del cliente SSH; *SSH Arguments*, el comando que ejecutará cuando establezca la conexión con la máquina virtual, por ejemplo: `ssh -i user1key ubuntu@172.31.9.22`, este es el mismo comando que se utiliza para conectarse a una máquina virtual utilizando la línea de comandos a través de las *euca2ools*; *SSH Key Template* y *EC2 Private Key Template* se establecen cuando se escoge la llave privada de *Putty* para la conexión con la máquina virtual; *SSH User*, es el usuario con el que se conectará a la máquina virtual; es importante distinguir si la máquina es de Ubuntu o de Centos, puesto que por defecto el usuario con el que se conecta a Ubuntu es "ubuntu" y el usuario de Centos es "root".

En la Figura 3.15 se muestra el botón que permite conectarse a la máquina virtual, utilizando los parámetros definidos en la ventana *Tools*.

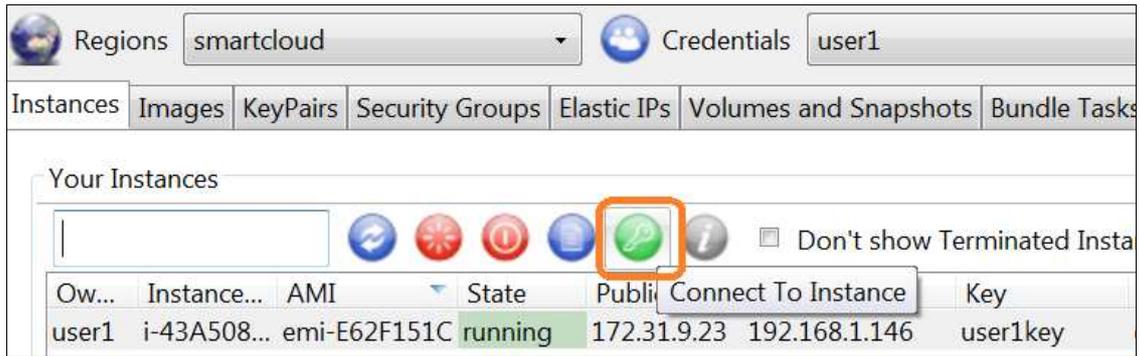


Figura 3.15 Botón para la conexión a una instancia desde *Elasticfox*.

Después de pedir el archivo que contiene la llave privada *Putty*, el programa ejecutará el cliente SSH (Figura 3.16) y finalmente se establecerá la conexión.

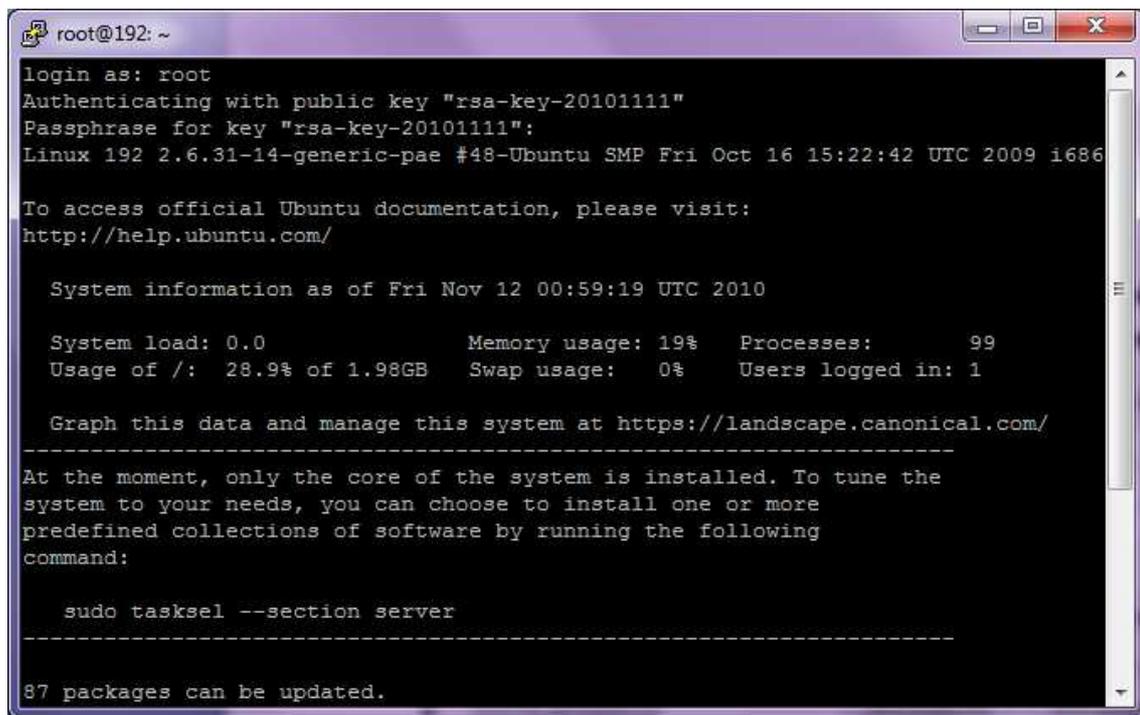


Figura 3.16 Inicio de sesión en una instancia desde *Elasticfox* a través de *Putty*

Ya que se utilizó una llave privada para el Inicio de sesión, no fue necesario establecer ninguna contraseña; si eso se requiere, basta con configurarla en el *Puttygen* cuando se convierte la llave privada de *Eucalyptus* en la llave privada de *Putty*.

3.3 ADMINISTRACIÓN DE VOLÚMENES

Un volumen es un tipo especial de dispositivo de bloque que permite acceder directamente a un dispositivo de almacenamiento como un disco duro, sin pasar por caché ni *buffers*⁷⁰ del sistema operativo. [83]

Un usuario puede utilizar un volumen de acuerdo a sus requerimientos, es decir, puede almacenar archivos, servicios o cualquier tipo de información para poder utilizarlos en una máquina virtual.

El componente que permite trabajar con volúmenes es el denominado *Storage Controller* (Sección 2.2.1.5).

A continuación se muestra el proceso necesario para la creación de un volumen lógico a través de línea de comandos y *Elasticfox*.

En el Código 3.1 se puede observar la creación de un volumen lógico para lo cual es necesario especificar el tamaño en (GB) y la zona donde se desea crearlo, en este caso se utilizó la zona llamada “smartcloud” correspondiente al *cluster* que se utilizó para este proyecto.

```
euca-create-volume --size 8 -z smartcloud
```

Código 3.1 Creación de un volumen línea de comandos

Para la creación de un nuevo volumen en la interfaz de *Elasticfox*, se escoge la opción “*Volumes and Snapshots*”; como se muestra en la Figura 3.17, se deben describir las características del volumen a crearse.

⁷⁰ **Buffer** es una ubicación de la memoria reservada para el almacenamiento temporal de información digital, mientras está esperando ser procesada.

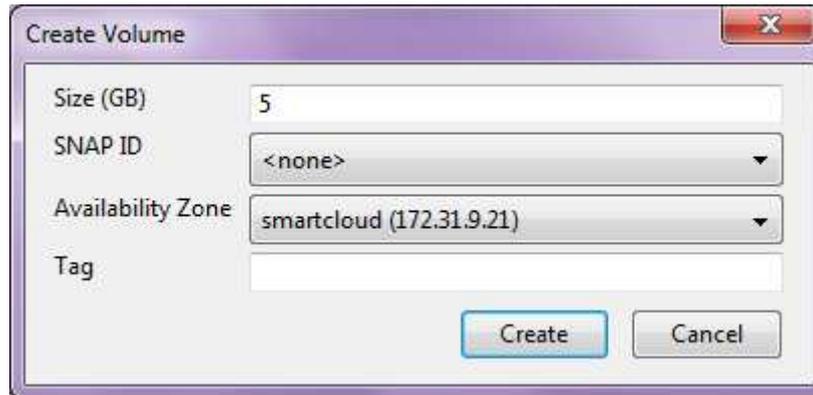


Figura 3.17 Creación de un volumen en *Elasticfox*

Para asociar un volumen a una instancia, ésta debe estar en ejecución, se selecciona dicha instancia y se recopilan ciertas características necesarias como: número de identificación de la instancia cuyo formato es similar a i-xxxxxxx, el cual está dentro de las especificaciones de la instancia en ejecución; identificación del volumen con el cual se la va a asociar (vol-xxxxxxx), que es información que se muestra en la sección “*volumes*” de *Elasticfox*, y el dispositivo con el que se desea asociar el volumen, este último representa una ubicación que el usuario le sugiere al sistema para el montaje del volumen puede ser el mismo sugerido siempre y cuando otro dispositivo no esté usando la misma ubicación. (Figura 3.18)

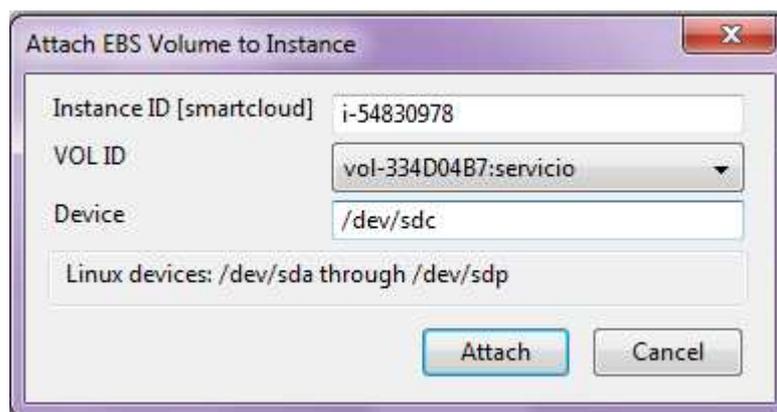


Figura 3.18 Asociación de un volumen a una instancia *Elasticfox*

El Código 3.2, muestra la asociación de un volumen a una instancia a través de la línea de comandos con las *euca2ools*.

```
euca-attach-volume -i i-4FEF0928 -d /dev/sdc vol-32AA04A6
```

Código 3.2 Asociación de un volumen a una instancia con la línea de comandos

La Figura 3.19 y Figura 3.20 muestran un volumen asociado a una instancia, utilizando la herramienta de gestión web (*Elasticfox*) y la línea de comandos (*Euca2ools*).

VOL ID	Size (GB)	SNAP ID	Availability Zone	Status	Local Creation Date	Instance ID	Device	Attachm...	Attach Time	Tag
vol-334D04B7	10		smartcloud	available	2010-11-13 09:16:46					servicio
vol-333804B5	8	snap-331A0...	smartcloud	available	2010-11-13 13:05:17					htservice2
vol-334304B9	10	snap-33840...	smartcloud	available	2010-11-16 18:36:12					
vol-32AA04A6	9		smartcloud	in-use	2010-11-21 12:20:54	i-4FEF0928	/dev/sdc	attaching	2010-11-21 13:...	lamp
vol-330104B0	8		smartcloud	available	2010-11-07 19:13:32					htservice
vol-336204B8	10	snap-334C0...	smartcloud	available	2010-11-13 12:02:22					prueba

Figura 3.19 Volumen asociado a una instancia *Elasticfox*

```
root@cloud:~# euca-describe-volumes
VOLUME vol-334D04B7      10      smartcloud      available      2010-11-13T14:16:46.12Z
VOLUME vol-333804B5     8      snap-331A04AB  smartcloud     available      2010-11-13T18:05:17.546Z
VOLUME vol-334304B9    10      snap-338404BE  smartcloud     available      2010-11-16T23:36:12.171Z
VOLUME vol-32AA04A6     9      smartcloud     in-use        2010-11-21T17:20:54.539Z
ATTACHMENT vol-32AA04A6     i-4FEF0928      unknown,requested:/dev/sdc      2010-11-21T21:01:53.605Z
VOLUME vol-330104B0     8      smartcloud     available      2010-11-08T00:13:32.352Z
VOLUME vol-336204B8    10      snap-334C04B8  smartcloud     available      2010-11-13T17:02:22.814Z
VOLUME vol-325804A1    10      smartcloud     available      2010-11-09T02:20:49.22Z
root@cloud:~#
```

Figura 3.20 Volumen asociado a una instancia *euca2ools*

3.4 IMPLEMENTACIÓN DE UN SERVICIO EN UNA INSTANCIA

Para comprobar la implementación de la persistencia de información de una máquina virtual, se implementó un servidor web y un motor de base de datos mysql, en ella para lo cual se utilizaron los métodos presentados en las secciones (3.4.1 y 3.4.2).

3.4.1 MÉTODO DE ENLACE DE ARCHIVOS

La idea detrás de este método, es tener la capacidad de ejecutar un servidor con los datos almacenados en los volúmenes lógicos, copiando los archivos del servicio al volumen y creando enlaces simbólicos a éste.

Se implementó un servidor web apache2 para la publicación de un sitio web haciendo uso de un volumen.

Para la configuración se debe tener acceso a una máquina virtual en ejecución a través de un cliente SSH, en este caso, se va a acceder a la máquina virtual que utiliza la imagen de Centos (Tomar como referencia la Figura 3.10).

Una vez que se tiene acceso a la máquina virtual con las credenciales de root, se procede a dar formato al volumen utilizando las herramientas “mkfs⁷¹”, con las cuales se le va a dar el formato de sistemas de archivos ext3. Adicionalmente se debe crear un directorio, en este caso /vol1, que luego se utiliza como punto de montaje del volumen.

Luego, para brindar el servicio, se debe instalar en la máquina virtual el software necesario, en este caso el paquete httpd.

⁷¹ **mkfs** son herramientas para dar formato, en este caso, a un volumen lógico con un sistema de archivos.

Una vez instalado el paquete requerido, el servicio httpd debe estar en ejecución. Es importante configurar el servicio de tal forma que se inicialice cada vez que la instancia arranque. (Código 3.3)

```
/sbin/chkconfig httpd on
```

Código 3.3 Configuración del servicio para que se inicialice cuando la instancia arranca

Para configurar el servicio y usar el volumen, lo primero que se hace es detener el servicio de apache. A continuación se copian los archivos del sitio web al volumen y se crea un enlace simbólico a éste. Como paso final, se inicializa el servicio de apache. (Código 3.4)

```
# Mover los archivos de httpd al volumen:
mv /etc/httpd /voll/httpd

# Crear un enlace simbolico al volumen:
ln -s /voll/httpd /etc

# Se mueve el contenido del directorio donde se aloja el sitio web y
# se crea un enlace simbolico:
mv /var/www /voll/www
ln -s /voll/www /var/
```

Código 3.4 Configuración de un volumen para el servidor web

Una vez que el servicio se está ejecutando en el volumen, se puede probar la funcionalidad de éste creando una nueva máquina virtual que se asocie con dicho volumen, dentro de dicha máquina virtual se crea el directorio donde se debe montar el volumen, y finalmente se inicializa el servicio. Esto prueba que la información generada en una máquina virtual, gracias a los volúmenes, puede ser utilizada en cualquier otra ya que se ha mantenido la información.

Todos los datos son almacenados de forma persistente en el volumen, es por eso que al adjuntar el volumen que posee dichos datos a una nueva instancia, ésta

puede ejecutar nuevamente el servicio con la información almacenada desde la máquina virtual antigua.

Finalmente se podrá observar el sitio web (Figura 3.21) publicado en /vol1/var/www. Para acceder al sitio se requiere utilizar la IP “pública” asignada a la máquina virtual a la cual está vinculado el volumen que contiene el sitio.



Figura 3.21 Sitio web publicado en vol1

Como una prueba adicional de lo anterior, se ha utilizado una máquina virtual de Ubuntu que maneja un volumen lógico como un mecanismo de almacenamiento persistente para un servidor de base de datos *mysql*; como se va a detallar más adelante, no se crean enlaces lógicos hacia el volumen, como se hizo en (Código 3.4), sino que se van a montar los archivos del servicio almacenados en el volumen en la máquina virtual.

Al igual que en la implementación anterior, se debe acceder a la máquina virtual en ejecución la cual se asocia a un volumen, una vez que se accede a la máquina virtual se debe instalar el software necesario para esta implementación (Código 3.5).

```
sudo apt-get update && sudo apt-get upgrade -y
sudo apt-get install -y xfsprogs72 mysql-server
```

Código 3.5 Instalación del servidor de base de datos *mysql*

Se crea un sistema de archivos de extensión XFS⁷³ o ext3 en el volumen y se lo monta en el directorio `/volmysql`. Para que *mysql* utilice el volumen, primero se debe detener el servicio, mover y luego montar los archivos del servicio de base de datos al volumen (Código 3.6); cabe recalcar que no se crean enlaces simbólicos de la máquina virtual al volumen como en la Sección 3.4.1.

```
echo "/vol/etc/mysql /etc/mysql none bind" | sudo tee -a /etc/fstab
sudo mount /etc/mysql

echo "/vol/lib/mysql /var/lib/mysql none bind" | sudo tee -a /etc/fstab
sudo mount /var/lib/mysql

echo "/vol/log/mysql /var/log/mysql none bind" | sudo tee -a /etc/fstab
sudo mount /var/log/mysql
```

Código 3.6 Montar los archivos que *mysql* requiere manejar en una instancia

Finalmente, se debe reiniciar el servicio *mysql*, y crear una base de datos con el comando “`mysql -u root -p -e 'CREATE DATABASE CLOUD'`” (Figura 3.22) para almacenarla y luego poderla utilizar en otra máquina virtual utilizando un volumen lógico o un *snapshot*; es necesario recordar que éste último guarda el estado actual de todos los archivos almacenados en el volumen, incluyendo la información de la base de datos de *mysql* y sus *logs* binarios.

⁷² **Xfsprogs** es un conjunto de herramientas para el manejo del sistema de archivos XFS.

⁷³ **XFS** es un sistema de archivos de 64 bits de alto rendimiento disponible para la plataforma UNIX. Más información en: <http://es.wikipedia.org/wiki/XFS>.

```

root@192: ~
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| CLOUD      |
| mysql      |
+-----+
3 rows in set (0.00 sec)

mysql>

```

Figura 3.22 Base de datos “CLOUD” creada

Para probar el almacenamiento de la base de datos, se puede crear una nueva máquina virtual con el software necesario para la ejecución del servicio, se monta y asocia el volumen a la máquina virtual, *mysql* debe apuntar a los archivos de base de datos correctos en el volumen. Actualmente se tiene una copia exacta de la base de datos que se creó al principio, lo que se puede verificar buscando dicha base de datos.

Si se desea restaurar una base de datos que ha sido respaldada con anterioridad, el usuario puede manejar otra máquina virtual, un volumen lógico y también un *snapshot* creado a partir de dicho volumen, es decir, una vez que una máquina virtual en ejecución almacena en el volumen lógico una base de datos creada en el gestor de base de datos *mysql*, el cliente puede hacer un respaldo de dicho volumen utilizando las herramientas de línea de comandos (Código 3.7) o a través de la herramienta de gestión *Elasticfox*, en otras palabras puede crear un *snapshot* del volumen que tiene almacenado la información y servicios del usuario (Figura 3.23).

```
euca-create-snapshot vol-329404A7
```

Código 3.7 Creación de un *snapshot* utilizando *euca2ools*

Saved Snapshots						
<input type="text"/> ⊕ ⊖ ⊗ ⊙						
No Filter ⌵						
SNAP ID	VOL ID	Status	Local Start Time	Progress	Tag	⊞
snap-32C604A8	vol-329404A7	completed	2010-12-10 13:29:50	100%	snapshot mysql	
snap-333904B8	vol-331804BA	completed	2010-11-27 15:05:58	100%		

Figura 3.23 Visualización en *Elasticfox* del *snapshot* creado

VOL ID	Size ...	SNAP ID	Status	Instance ID	Device	Attachment Status	Tag
vol-336204B8	10	snap-334C04B8	available				
vol-330804B9	7		in-use	i-4C1E0881	unknown,requested:/dev/sdb	attached	
vol-333304B6	7	snap-333904B8	in-use	i-4C8108A0	unknown,requested:/dev/sdb	attached	
vol-331804BA	7		in-use	i-3EE407ED	unknown,requested:/dev/sdb	attached	
vol-332B04B1	7	snap-333904B8	in-use	i-505209A2	unknown,requested:/dev/sdb	attached	
vol-32EA04AC	7	snap-333904B8	in-use	i-4B3409AB	unknown,requested:/dev/sdb	attached	
vol-334304BE	7	snap-333904B8	in-use	i-502E08D8	unknown,requested:/dev/sdb	attached	
vol-326704A6	7		available				
vol-332604B2	7	snap-333904B8	in-use	i-45FD08A0	unknown,requested:/dev/sdb	attached	
vol-32B604AD	10	snap-32C604A8	available				vol snap mysql
vol-332704B0	7	snap-333904B8	in-use	i-4C1E0881	unknown,requested:/dev/sdc	attached	

Figura 3.24 Nuevo volumen creado a partir de un *snapshot*

Para comprobar la funcionalidad del *snapshot*, a partir del mismo, se debe crear un nuevo volumen lógico, que se asocie a otra máquina virtual en ejecución, y en ella poder probar si la base de datos almacenada con anterioridad está correctamente guardada. El cliente no debe olvidar crear un punto de montaje para el nuevo volumen en la instancia; cabe recalcar que ya no se le debe dar formato al volumen.

Para mayor seguridad, al crear la base de datos original, se puede crear un usuario con su respectiva contraseña para poder acceder a ésta, por lo que en esta nueva máquina virtual sería necesario cambiar el propietario y grupo al que pertenece `/mysql` (Código 3.8), a continuación se detiene el servicio y se monta los archivos de *mysql* (Código 3.6).

```
sudo find /vol/{lib,log}/mysql/ ! -user root -print0 |
#Cambiar el propietario de
  sudo xargs -0 -r chown mysql
sudo find /vol/{lib,log}/mysql/ ! -group root -a ! -group adm -print0
|
  sudo xargs -0 -r chgrp mysql
```

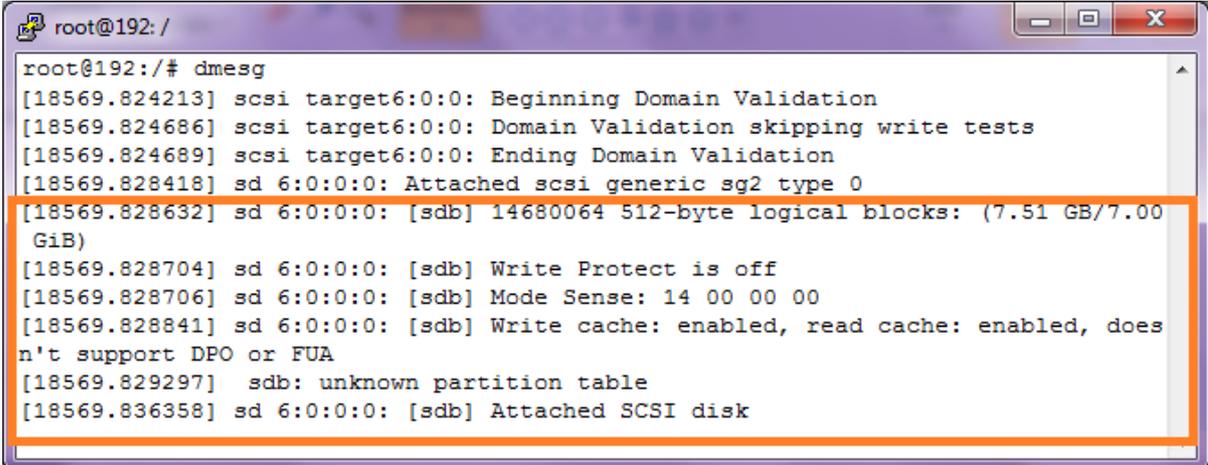
Código 3.8 Cambio de propietario y grupo al que pertenece *mysql*

Finalmente, se inicializa el servicio y verifica que la base de datos creada anteriormente se encuentra en esta instancia (`mysql -u root -p -e 'SHOW DATABASES'`).

3.4.2 MÉTODO *CHROOT*

Este método representa otra alternativa para mantener persistente el sistema de archivos completo. Para tal propósito se instala un sistema operativo básico en el volumen y se utiliza *chroot*⁷⁴ para instalar y ejecutar el servicio requerido.

Una vez que el volumen esté vinculado a la instancia en la que se requiere que se ejecute el servicio, es necesario determinar el dispositivo asociado al volumen que detectará el sistema operativo en el momento de la vinculación, para lo cual se puede utilizar el comando “*dmesg*”⁷⁵ (Figura 3.25).



```

root@192:/# dmesg
[18569.824213] scsi target6:0:0: Beginning Domain Validation
[18569.824686] scsi target6:0:0: Domain Validation skipping write tests
[18569.824689] scsi target6:0:0: Ending Domain Validation
[18569.828418] sd 6:0:0:0: Attached scsi generic sg2 type 0
[18569.828632] sd 6:0:0:0: [sdb] 14680064 512-byte logical blocks: (7.51 GB/7.00 GiB)
[18569.828704] sd 6:0:0:0: [sdb] Write Protect is off
[18569.828706] sd 6:0:0:0: [sdb] Mode Sense: 14 00 00 00
[18569.828841] sd 6:0:0:0: [sdb] Write cache: enabled, read cache: enabled, does n't support DPO or FUA
[18569.829297] sdb: unknown partition table
[18569.836358] sd 6:0:0:0: [sdb] Attached SCSI disk

```

Figura 3.25 Volumen reconocido como dispositivo en el sistema operativo

La máquina virtual que se utiliza con este método fue creada utilizando la imagen correspondiente a Ubuntu.

Después de que el volumen (/dev/sdb) ha sido asociado, se le da formato utilizando el sistema de archivos ext3 y se crea un punto de montaje (/lapsevice) para que después pueda ser montado. El siguiente paso es instalar un sistema

⁷⁴ **Chroot** es una llamada al sistema en UNIX que permite configurar un directorio como "raíz" del sistema de ficheros para un proceso y sus hijos. Permite configurar el sistema de forma tal que se puedan lanzar procesos confinados dentro de un determinado directorio.

⁷⁵ **dmesg** (diagnostic message) es un comando presente en los sistemas operativos Unix que lista el buffer de mensajes del kernel.

operativo (Ubuntu) básico utilizando el comando *debootstrap*⁷⁶, en el directorio `/usr/share/debootstrap/scripts`, se encuentran todas las distribuciones disponibles, en este caso se utilizó “*lucid*”⁷⁷. Se montan los directorios `/proc`, `/sysfs` y `/dev` para poder instalar el servicio a través de *chroot*, para lo cual se debe haber actualizado el nuevo sistema operativo y haber configurado la hora local del sistema utilizando *tzdata*. Todo este proceso se lo presenta en el Código 3.9.

```
mkfs.ext3 /dev/sdb
mount /dev/sdb /lapsevice
apt-get install debootstrap
debootstrap lucid /lapsevice
mount proc /lapsevice /proc -t proc
mount sysfs /lapsevice /sys -t sysfs
mount -o bind /dev /lapsevice/dev
chroot /lapsevice
apt-get update
dpkg-reconfigure tzdata
```

Código 3.9 Instalación de un nuevo sistema operativo básico en un volumen lógico

Se instala el nuevo servidor (*apache2*) se copia el sitio web, previamente creado, en el directorio `/var/www` y se reinicia el servicio (Código 3.10). [85]

El sitio web se copió desde un dispositivo de almacenamiento en red, *NAS*⁷⁸ utilizando la herramienta *smbclient*⁷⁹ ya que el *NAS* es también un servidor *samba* el cual permite compartir archivos entre sistemas *Windows* y *Unix*.

```
apt-get install apache2
apt-get install php5 libapache2-mod-php5
/etc/init.d/apache2 restart
```

Código 3.10 Instalación del servidor web en el volumen lógico

⁷⁶ **debootstrap** instala algunos paquetes esenciales en algún directorio para su uso con *chroot*. Los paquetes esenciales no implican en sí el sistema operativo. Se puede usar *apt-get* para instalar otros paquetes que hace que *Ubuntu* sea utilizable. [84]

⁷⁷ **Lucid** Linux Ubuntu 10.04.

⁷⁸ **NAS** (*Network Attached Storage*) es una tecnología de almacenamiento dedicada a compartir la capacidad de almacenamiento de un computador con ordenadores clientes a través de una red, haciendo uso de un Sistema Operativo optimizado para dar acceso con los protocolos *CIFS*, *NFS*, *FTP* o *TFTP*.

⁷⁹ **Smbclient** es una herramienta que permite intercambiar ficheros con comandos como *get* y *put* con servidores *SMB* (*samba*).

Si la máquina virtual se apaga, todo el sistema de archivos se borra, pero el sistema operativo básico instalado en el volumen, persistirá, así que si se requiere utilizar el mismo volumen en otra máquina virtual, bastará con montar (proc, sysfs y dev) en la nueva máquina virtual.

Finalmente, se podrá observar el sitio web (Figura 3.26) publicado en /lapsevice/var/www. Para acceder al sitio se requiere utilizar la IP “pública” asignada a la máquina virtual a la cual está vinculada el volumen que contiene el sitio.



Figura 3.26 Sitio web publicado en un volumen lógico con *chroot*

Estos métodos son algunas de las opciones que existen para permitir el almacenamiento persistente de los datos en *Eucalyptus*, es el usuario quien decide cuál de ellos utilizar de acuerdo a sus requerimientos.

3.5 PRESTACIONES PARA LOS USUARIOS

Considerando la configuración establecida en el archivo “eucalyptus.conf”, la red que se utilizó para la interfaz privada de las máquinas virtuales fue **192.168.1.0** con máscara **255.255.255.0** de tal forma que en la Tabla 3.1 se muestran los resultados obtenidos de los cálculos realizados para definir el número máximo de usuarios utilizando esta configuración.

Se ha realizado una división entre el número de hosts de la dirección de red y el máximo número de direcciones IP por red (VNET_ADDRESSPERNET) para poder obtener el número de redes simultáneas activas (grupos de seguridad activos), el número de direcciones válidas por red corresponde al número máximo de direcciones IP que podrán ser asignadas a las máquinas virtuales de un usuario, considerando que una dirección es para red y la otra es para *broadcast*. El máximo número de usuarios para esta configuración es el resultado de la multiplicación entre Redes Simultáneas Activas y Direcciones Válidas por Red ($16 \times 14 = 224$), considerando los nodos que actualmente se tienen disponibles, se debe conocer que entre ambos se puede correr como máximo 20 máquinas virtuales, y si se divide el Máximo de Usuarios para el Número de Máquinas que puede ejecutar el nodo1(diez y seis) se tendrá como resultado el Número Máximo de Nodos de las mismas características ($208/16 = 13$). Si se utilizaran solamente nodos de las características del nodo 2 (cuatro MVs) entonces se necesitarán 52 equipos.

máximo de hosts	256
máximo de direcciones por red	16
Redes Simultáneas Activas	16
Direcciones Válidas por red	14
Máximo de usuarios de acuerdo a esta configuración	224
Número de nodos necesarios (características nodo 1)	14
Número de nodos necesarios (características nodo 2)	56

Tabla 3.1 Cálculo del máximo de usuarios con la configuración establecida

Dentro de esta configuración (eucalyptus.conf en el *front-end*) se pueden tener hasta 208 usuarios cada uno ejecutando una máquina virtual (Tabla 3.1). Sin

embargo, si se requiere contar con más usuarios, lo recomendable sería utilizar en equipos separados los componentes del *front-end* como lo que se especificó en la Sección 2.2.1 y además utilizar otra red privada, una clase B o A, y establecer en VNET_ADDRESSPERNET del archivo eucalyptus.conf el valor de direcciones requerido.

El número de nodos controlados por un CC, dependerá de la configuración en el archivo eucalyptus.conf, si se requiere escalar aun más el sistema, se deberá utilizar varios CC con varios NCs y establecer una configuración de acuerdo a los requerimientos.

En este prototipo se utilizaron dos nodos, el nodo 1 soporta hasta 16 máquinas virtuales y el nodo 2 soporta hasta 4 de tamaño m1.small o c1.medium, ejecutándose simultáneamente; por lo tanto el prototipo como tal puede soportar máximo hasta 20 instancias en ejecución. Pero se pueden aumentar más con la variable MAX_CORES en el archivo de configuración de los nodos sin embargo, hay que tomar en cuenta el rendimiento de CPU. Se recomienda que se aumente la capacidad de ejecución de máquinas virtuales hasta 16 para que el rendimiento no se vea tan afectado; las pruebas que se realizaron se muestran en la Sección 3.7 de este capítulo.

Por lo tanto existen dos formas de escalar la capacidad en una Nube, aumentando el número de nodos o aumentando el valor de MAX_CORES en el archivo eucalyptus.conf de cada nodo, pero tomando en cuenta la disminución de rendimiento en el CPU de cada nodo.

3.6 VELOCIDAD DE TRANSMISIÓN Y TRÁFICO DEL PROTOTIPO

Para trabajar con aplicaciones alojadas en una Nube, lo recomendable es utilizar una conexión de alta velocidad, de esta forma se garantiza un buen flujo de datos sin obstrucciones ni cortes. Por tal razón, se ha utilizado el programa *iptraf* para determinar la velocidad necesaria para establecer una conexión confiable.

lptraf es una herramienta basada en consola para mostrar estadísticas de red, monitoreo de tráfico y paquetes de red a través de TCP y UDP [90].

En este caso la visualización de los datos de la velocidad de transmisión se realizó en la interfaz *eth1* del *front-end*, que es la interfaz a través de la cual los clientes acceden al servicio que ofrece la Nube.

Se utilizó el número máximo de máquinas virtuales ejecutándose al mismo tiempo, a cada una de ellas se le vinculó un volumen lógico que contiene un sitio web publicado, de esta manera se establecieron conexiones simultáneas a las instancias que tienen direcciones IP públicas (dentro de la red de la EPN).

La medición de la velocidad se realizó en los siguientes casos:

- a) Conexión de un usuario con la Nube a través de *Elasticfox* para la ejecución de las ocho máquinas virtuales (Figura 3.27).

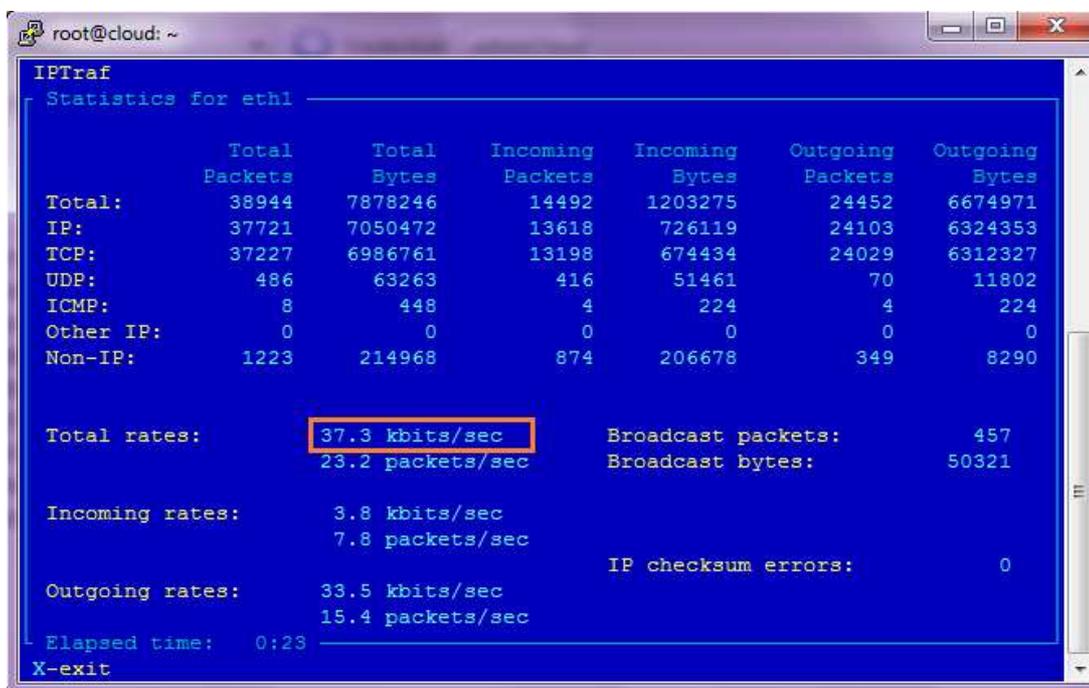


Figura 3.27 Resultado de *iptraf* en *eth1* (8 mvs en ejecución)

- b) La Figura 3.28 muestra la velocidad de transmisión en la interfaz *eth1* cuando todas las máquinas virtuales, que el prototipo ofrece a los clientes, están en

ejecución, y los usuarios tienen una conexión con sus respectivas máquinas virtuales a través de un cliente SSH (putty) para la creación y montaje de volúmenes lógicos.

```

IPTraf
- Statistics for eth1
-
      Total      Total      Incoming      Incoming      Outgoing      Outgoing
      Packets    Bytes     Packets     Bytes     Packets     Bytes
Total:      53056    10890249    19896     1627939    33160     9262310
IP:         51410    9764345    18717     976517    32693     8787828
TCP:        50615    9669717    18084     901416    32531     8768301
UDP:         795      94628      633       75101     162       19527
ICMP:        0         0          0          0          0          0
Other IP:    0         0          0          0          0          0
Non-IP:     1646     288908    1179      278666    467       10242

Total rates:      41.5 kbits/sec      Broadcast packets:      602
                  25.8 packets/sec      Broadcast bytes:       64348

Incoming rates:   4.9 kbits/sec
                  9.2 packets/sec

Outgoing rates:   36.6 kbits/sec
                  16.6 packets/sec

IP checksum errors: 0

Elapsed time:    0:30
X-exit
  
```

Figura 3.28 Resultados de *iptraf* en la interfaz eth1 con la conexión de usuarios a todas las máquinas virtuales en ejecución

- c) Conexión a dos máquinas virtuales que tienen IPs públicas (red EPN) para descargar y visualizar el contenido de sus respectivos sitios web. Esta prueba se realizó con 10 usuarios en cada máquina virtual.

```

IPTraf
Statistics for eth1

      Total      Total      Incoming      Incoming      Outgoing      Outgoing
      Packets    Bytes      Packets      Bytes      Packets      Bytes
Total:    44844    9087400    16646    1376875    28198    7710525
IP:       43411    8129296    15622    823137    27789    7306159
TCP:     42880    8060977    15165    766844    27715    7294133
UDP:       523     67871     453     56069     70     11802
ICMP:        8      448      4      224      4      224
Other IP:   0      0      0      0      0      0
Non-IP:   1433    252258    1024    242664     409     9594

Total rates:    57.2 kbits/sec      Broadcast packets:    520
                35.4 packets/sec      Broadcast bytes:    57299

Incoming rates: 18.4 kbits/sec
                15.0 packets/sec

Outgoing rates: 38.7 kbits/sec
                20.4 packets/sec

IP checksum errors: 0

Elapsed time: 0:26
X-exit

```

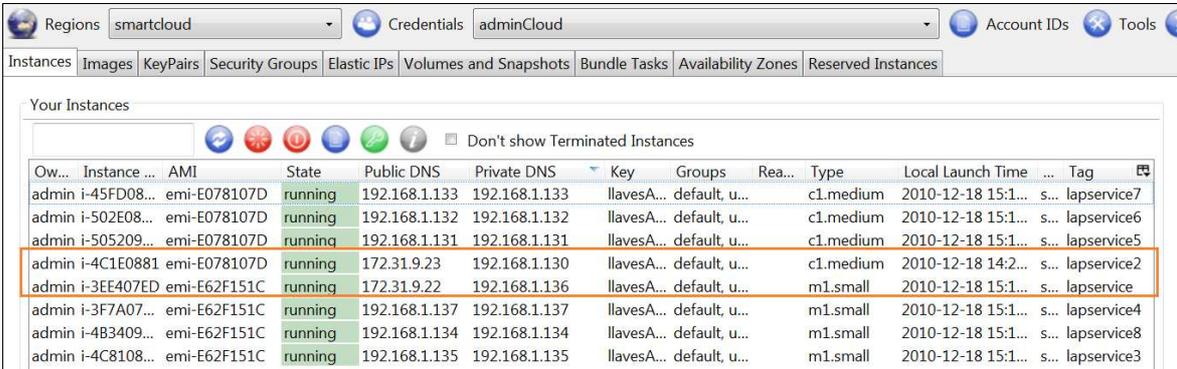
Figura 3.29 Resultados de *iptraf* en la interfaz eth1 con todas las máquinas virtuales en ejecución asociadas a volúmenes lógicos

Los resultados que se pueden visualizar, son resultados estimados considerando que se pueden tener más máquinas virtuales con IPs públicas que permitirán aceptar un gran número de usuarios para la visualización del sitio web publicado. También se debe tomar en cuenta el tipo de aplicaciones que se van a instalar en una máquina virtual, para lo cual se tendrá que realizar un análisis para dichos casos. Si la velocidad no fuera suficiente, se tendrá entonces, que aumentarla para mantener la calidad del servicio.

La calidad del servicio en cuanto a velocidad de transmisión puede ser sobrellevada utilizando una conexión de mayor velocidad; en cuanto a rendimiento de máquinas virtuales dentro del prototipo, si se utilizan (máximo 2) máquinas virtuales de tamaño c1.xlarge ó (máximo 8) máquinas virtuales de tamaño m1.small, se estará utilizando la misma cantidad de recursos que puede ofrecer la Nube; en cuanto a disponibilidad se puede aumentar el número de nodos para ofrecer mayor cantidad de máquinas virtuales ejecutándose simultáneamente.

Para monitorear el tráfico en la interfaz de comunicación externa (eth1), se utilizó la herramienta *iftop*.

Iftop es una herramienta para monitorear el tráfico y mostrar el uso del ancho de banda de una red. Los resultados del análisis son entregados en tiempo real. Muestra la actividad entre un par de *hosts*, incluyendo picos de tráfico a los 40 segundos, tráfico total transferido/recibido y velocidad de transferencia promedio a los 2, 10 y 40 segundos [91].



Ow...	Instance ...	AMI	State	Public DNS	Private DNS	Key	Groups	Rea...	Type	Local Launch Time	...	Tag
admin	i-45FD08...	emi-E078107D	running	192.168.1.133	192.168.1.133	llavesA...	default, u...		c1.medium	2010-12-18 15:1...	s...	lapservice7
admin	i-502E08...	emi-E078107D	running	192.168.1.132	192.168.1.132	llavesA...	default, u...		c1.medium	2010-12-18 15:1...	s...	lapservice6
admin	i-505209...	emi-E078107D	running	192.168.1.131	192.168.1.131	llavesA...	default, u...		c1.medium	2010-12-18 15:1...	s...	lapservice5
admin	i-4C1E0881	emi-E078107D	running	172.31.9.23	192.168.1.130	llavesA...	default, u...		c1.medium	2010-12-18 14:2...	s...	lapservice2
admin	i-3FE407ED	emi-E62F151C	running	172.31.9.22	192.168.1.136	llavesA...	default, u...		m1.small	2010-12-18 15:1...	s...	lapservice
admin	i-3F7A07...	emi-E62F151C	running	192.168.1.137	192.168.1.137	llavesA...	default, u...		m1.small	2010-12-18 15:1...	s...	lapservice4
admin	i-4B3409...	emi-E62F151C	running	192.168.1.134	192.168.1.134	llavesA...	default, u...		m1.small	2010-12-18 15:1...	s...	lapservice8
admin	i-4C8108...	emi-E62F151C	running	192.168.1.135	192.168.1.135	llavesA...	default, u...		m1.small	2010-12-18 15:1...	s...	lapservice3

Figura 3.29 Máquinas Virtuales con direcciones “públicas”

La Figura 3.29 muestra las dos máquinas virtuales con direcciones IP en la red de la EPN, a través de esta red y usando NAT, pueden acceder a Internet.

A continuación se muestran los resultados de la utilización de *iftop*. El comando que se utilizó fue (*iftop -i eth1 -P*). Donde la opción *-i* permite escoger la interfaz en la cual se requiere el monitoreo y *-P* muestra el puerto en el cual se estableció la conexión.

La Figura 3.30 muestra el tráfico en la interfaz eth1, se puede observar el tráfico SSH (22) y WWW (80) ya que son los puertos a través de los cuales se han establecido conexiones con *putty* y un navegador web respectivamente.

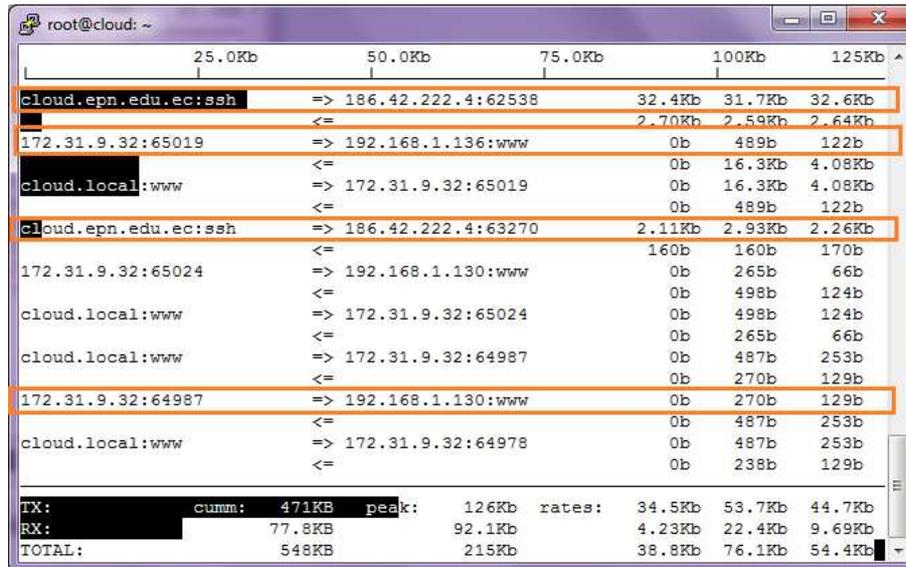


Figura 3.30 Resultado de *iftop*: tráfico de SSH y WWW a través de eth1

Las direcciones 192.168.1.130 y 192.168.1.136 corresponden a las máquinas virtuales con dos direcciones (Figura 3.29) una pública y una privada, en ambas máquinas virtuales se ha publicado un sitio web, es por tal razón que el tráfico mostrado en *iftop* corresponde a WWW.

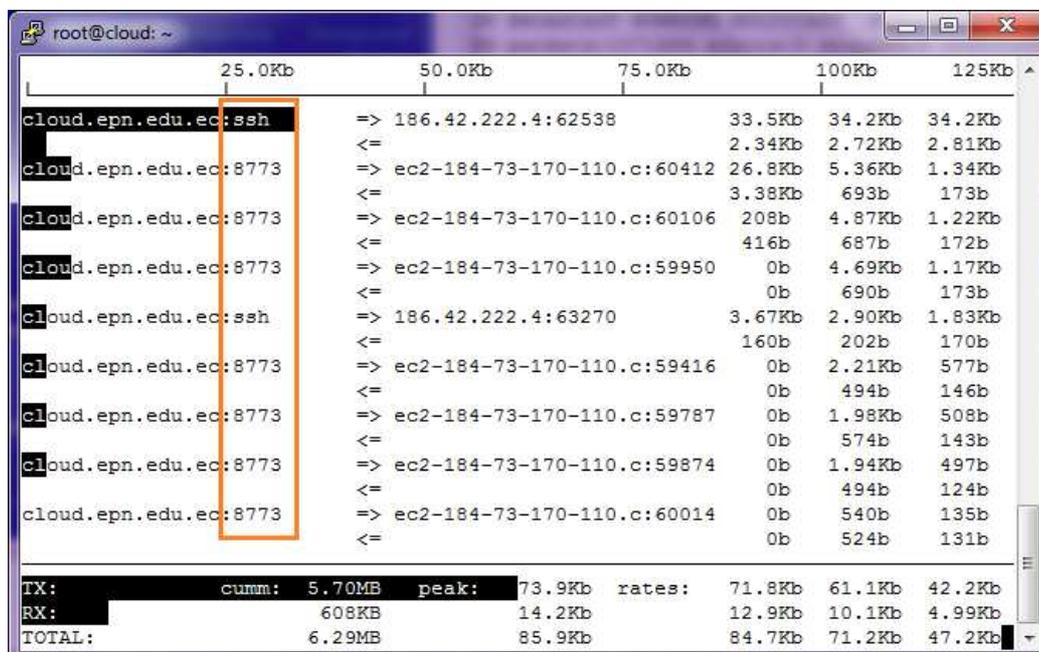


Figura 3.31 Resultado de *iftop*: muestra la conexión establecida entre Elasticfox y la Nube en el puerto 8773

En la Figura 3.31 se muestra la conexión entre *Elasticfox* (8 máquinas virtuales en ejecución) y la Nube (cloud.epn.edu.ec) en el puerto 8773 el cual se estableció en la configuración de *Elasticfox* en la Sección 2.5.8.

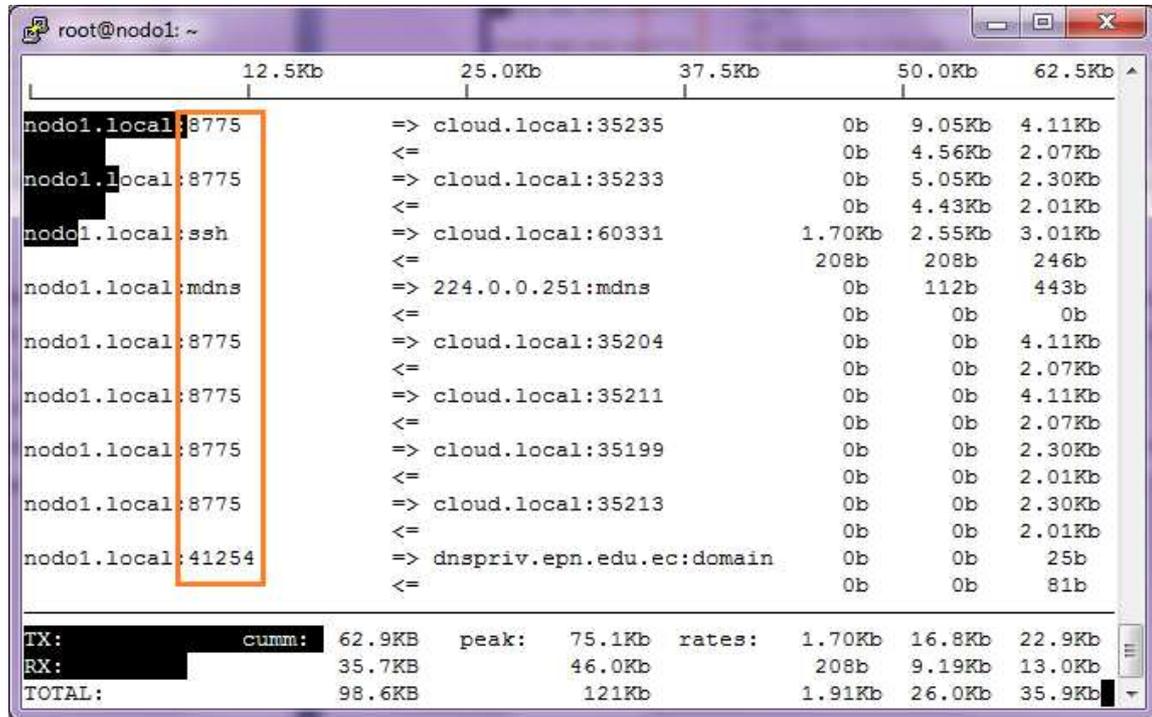


Figura 3.32 Resultado de *iftop* en *eth0* del *nodo1* que muestra el tráfico entre el *Cluster Controller* y el *Node Controller* en el puerto 8775

La Figura 3.32 muestra el tráfico entre el *Cluster Controller* y el *Node Controller* a través del puerto 8775 que se especificó en el archivo de configuración *eucalyptus.conf* del *Nodo1* (Código 2.8, Capítulo 2).

```

IPTraff
- Statistics for eth0 -
      Total      Total      Incoming      Incoming      Outgoing      Outgoing
      Packets    Bytes     Packets     Bytes     Packets     Bytes
Total:      53753    14168854    27836     3924081    25917     10244773
IP:         53753    13416312    27836     3534377    25917     9881935
TCP:        53520    13384999    27612     3506016    25908     9878983
UDP:         233      31313      224        28361      9          2952
ICMP:         0          0          0          0          0          0
Other IP:    0          0          0          0          0          0
Non-IP:      0          0          0          0          0          0

Total rates:      96.8 kbits/sec      Broadcast packets:      172
                  44.0 packets/sec      Broadcast bytes:      23847

Incoming rates:   27.8 kbits/sec
                  22.8 packets/sec

Outgoing rates:   69.0 kbits/sec
                  21.2 packets/sec

IP checksum errors:      0

Elapsed time:    0:21
X-exit

```

Figura 3.33 Resultado de *iptraf* que muestra la velocidad de transmisión en la interfaz eth0

La Figura 3.33 muestra la velocidad de transmisión en la interfaz eth0 (br0) del nodo1. Ya que los nodos son los encargados de la ejecución de las máquinas virtuales el tráfico generado en la red es alto, por lo tanto es muy recomendable utilizar una conexión Gigabit Ethernet (1000 Mbps). Dentro del prototipo la conexión Fast Ethernet (100 Mbps) es suficiente.

Con los resultados obtenidos en el *front-end* se puede concluir que para el número de usuarios que actualmente soporta el prototipo una conexión Fast Ethernet (100 Mbps) de velocidad es suficiente para la conexión; sin embargo, si se requiere escalar el sistema, se recomiendan los valores especificados en la Sección 2.5.1.

3.7 USO DE CPU EN LOS NODOS QUE FORMAN PARTE DEL PROTOTIPO

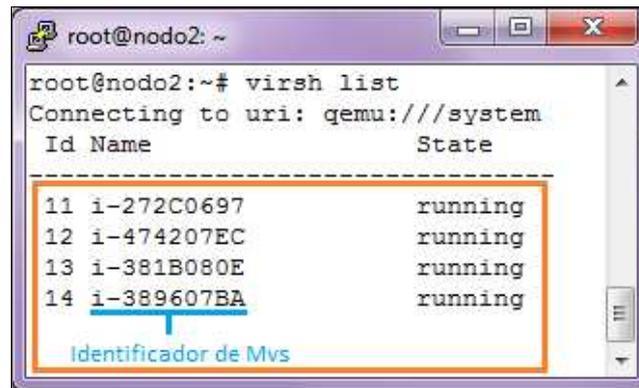
Para el presente proyecto el rendimiento de la Nube es una característica de suma importancia, en especial el rendimiento de los nodos que forman parte del prototipo, ya que éstos son los que ejecutan las máquinas virtuales.

El análisis del rendimiento de los nodos se ha realizado en base al uso del CPU de cada uno de éstos evaluado en diferentes casos, para lo cual se ha utilizado la herramienta *top*, que permite monitorear el sistema en tiempo real y de forma frecuente ya que publica una lista actualizada de los procesos que se están ejecutando.

Utilizando *top* se va visualizar los resultados de uso de CPU ejecutando MVs del tipo m1.small, c1.medium y c1.xlarge en los siguientes casos:

- a) El usuario Administrador ejecuta 4 máquinas virtuales en el nodo 2, cada una de tipo c1.medium, es decir, utiliza 1 CPU, 256 MB de memoria y 5 GB de disco; dichas MVs están listadas en la Figura 3.34 al utilizar el comando “virsh list”.

Una vez que se ejecuta el comando *top*, se despliega información individual de cada uno de los CPUs que forman parte del nodo 2 (Figura 3.35), de acuerdo a los resultados mostrados en dicha figura, el nodo 2 está formado por 4 CPUs, la primera columna del primer recuadro muestra el porcentaje de uso de CPU a nivel de usuario, se puede notar que al ejecutar cada una de las máquinas virtuales este porcentaje aumenta a 19.7%, 17.0%, 26.0% y 26.3%, ya que procesos “kvm” con prioridad PR=20, están siendo ejecutados por el usuario *root* cada vez que se arranca una máquina virtual, como se visualiza en el segundo recuadro. De esta forma el “%id”, tiempo en que la CPU ha estado sin trabajo y ha esperado ejecutar procesos de I/O (Entrada/Salida), también varía a 78.3%, 79.7%, 76.3% y 76.3% correspondiente a cada CPU.

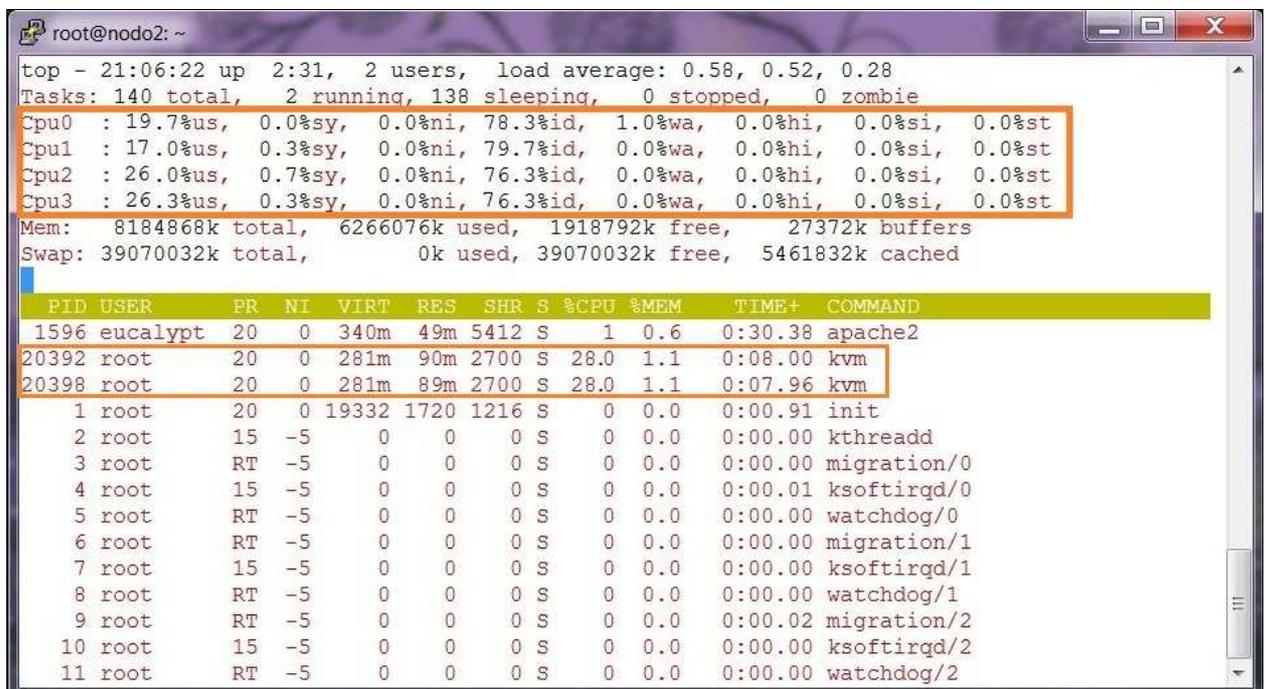


```

root@nodo2: ~
root@nodo2:~# virsh list
Connecting to uri: qemu:///system
 Id Name                               State
-----
 11 i-272C0697                          running
 12 i-474207EC                          running
 13 i-381B080E                          running
 14 i-389607BA                          running
  
```

Identificador de Mvs

Figura 3.34 Listado de MVs ejecutándose en el nodo 2



```

top - 21:06:22 up 2:31, 2 users, load average: 0.58, 0.52, 0.28
Tasks: 140 total, 2 running, 138 sleeping, 0 stopped, 0 zombie
Cpu0  : 19.7%us, 0.0%sy, 0.0%ni, 78.3%id, 1.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu1  : 17.0%us, 0.3%sy, 0.0%ni, 79.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu2  : 26.0%us, 0.7%sy, 0.0%ni, 76.3%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu3  : 26.3%us, 0.3%sy, 0.0%ni, 76.3%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem:   8184868k total, 6266076k used, 1918792k free, 27372k buffers
Swap: 39070032k total, 0k used, 39070032k free, 5461832k cached

  PID USER      PR  NI  VIRT  RES  SHR  S %CPU %MEM    TIME+  COMMAND
 1596 eucalypt  20   0  340m  49m 5412  S   1  0.6   0:30.38 apache2
20392 root      20   0  281m  90m 2700  S  28.0  1.1   0:08.00 kvm
20398 root      20   0  281m  89m 2700  S  28.0  1.1   0:07.96 kvm
   1 root      20   0 19332  1720 1216  S   0  0.0   0:00.91 init
   2 root      15  -5     0     0     0  S   0  0.0   0:00.00 kthreadd
   3 root      RT  -5     0     0     0  S   0  0.0   0:00.00 migration/0
   4 root      15  -5     0     0     0  S   0  0.0   0:00.01 ksoftirqd/0
   5 root      RT  -5     0     0     0  S   0  0.0   0:00.00 watchdog/0
   6 root      RT  -5     0     0     0  S   0  0.0   0:00.00 migration/1
   7 root      15  -5     0     0     0  S   0  0.0   0:00.00 ksoftirqd/1
   8 root      RT  -5     0     0     0  S   0  0.0   0:00.00 watchdog/1
   9 root      RT  -5     0     0     0  S   0  0.0   0:00.02 migration/2
  10 root      15  -5     0     0     0  S   0  0.0   0:00.00 ksoftirqd/2
  11 root      RT  -5     0     0     0  S   0  0.0   0:00.00 watchdog/2
  
```

Figura 3.35 Uso de CPU cuando 4 MVs c1.medium arrancan en el nodo 2

En la Figura 3.36 se visualiza un aumento en los valores de uso de CPU a nivel de usuario y una disminución de “%id”, ya que en cada una de las MVs se ha revisado el contenido de los repositorios y se ha instalado cualquier actualización de éstos, lo que implica mayor procesamiento y uso de la red, es decir el tiempo de espera de la CPU para la ejecución de operaciones de entrada y salida ha aumentado, ya que la CPU se altera entre los procesos y la velocidad a la que se ejecuta cada proceso no es uniforme.

```

root@nodo2: ~
top - 23:06:12 up 2 days, 4:31, 3 users, load average: 0.31, 0.32, 0.26
Tasks: 143 total, 2 running, 140 sleeping, 1 stopped, 0 zombie
Cpu0  : 18.6%us, 2.6%sy, 0.0%ni, 25.1%id, 53.1%wa, 0.3%hi, 0.3%si, 0.0%st
Cpu1  : 31.0%us, 0.0%sy, 0.0%ni, 67.9%id, 11.1%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu2  : 31.3%us, 1.0%sy, 0.0%ni, 67.8%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu3  : 32.8%us, 1.9%sy, 0.0%ni, 65.3%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem:   8184868k total, 7627888k used, 556980k free, 141752k buffers
Swap: 39070032k total, 0k used, 39070032k free, 6030508k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 21686 root        20   0  419m 219m 2700 S   36   2.7   10:18.50 kvm
 21635 root        20   0  410m 216m 2700 S   33   2.7    6:55.77 kvm
 21692 root        20   0  409m 219m 2700 S   22   2.7    5:46.11 kvm
 21670 root        20   0  419m 224m 2700 S    0   2.8    5:53.58 kvm
29437 root        20   0 19140 1336  988 R    0   0.0    0:00.39 top
   1 root        20   0 19332 1720 1216 S    0   0.0    0:00.91 init
   2 root        15  -5     0     0     0 S    0   0.0    0:00.00 kthreadd
   3 root         RT  -5     0     0     0 S    0   0.0    0:00.04 migration/0
   4 root        15  -5     0     0     0 S    0   0.0    0:00.22 ksoftirqd/0
   5 root         RT  -5     0     0     0 S    0   0.0    0:00.00 watchdog/0
   6 root         RT  -5     0     0     0 S    0   0.0    0:00.01 migration/1
   7 root        15  -5     0     0     0 S    0   0.0    0:00.03 ksoftirqd/1
   8 root         RT  -5     0     0     0 S    0   0.0    0:00.00 watchdog/1
   9 root         RT  -5     0     0     0 S    0   0.0    0:01.39 migration/2

```

Figura 3.36 Uso de CPU cuando las 4 MVs c1.medium actualizan sus repositorios

- b) Se ejecutan 4 máquinas virtuales en el nodo 2, cada una de tipo m1.small, es decir, utiliza 1 CPU, 128 MB de memoria y 2 GB de disco (Revisar Figura 2.41).

En este caso (Figura 3.37) se puede observar el uso total de los 4 CPUs, al ejecutar la misma cantidad de máquinas virtuales que en el ejemplo a) (c1.medium), pero esta vez, de capacidades menores (m1.small), y actualizar cada uno de sus repositorios. En comparación con el caso anterior, se nota que el uso de CPU total a nivel de usuario es menor, hablando del 29%, por lo tanto el “%id” no es del 100% sino del 77.3%.

```

root@nodo2: ~
top - 20:14:02 up 1:39, 2 users, load average: 0.14, 0.17, 0.17
Tasks: 140 total, 1 running, 139 sleeping, 0 stopped, 0 zombie
Cpu(s): 29.0%us, 0.3%sy, 0.0%ni, 77.3%id, 66.3%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 8184868k total, 6699240k used, 1485628k free, 21776k buffers
Swap: 39070032k total, 0k used, 39070032k free, 5828800k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 13894 root        20   0 19140 1348  988  S   0  0.0   0:00.37 top
     1 root        20   0 19332 1720 1216  S   0  0.0   0:00.91 init
     2 root        15  -5     0     0     0  S   0  0.0   0:00.00 kthreadd
     3 root        RT  -5     0     0     0  S   0  0.0   0:00.00 migration/0
     4 root        15  -5     0     0     0  S   0  0.0   0:00.01 ksoftirqd/0
     5 root        RT  -5     0     0     0  S   0  0.0   0:00.00 watchdog/0
     6 root        RT  -5     0     0     0  S   0  0.0   0:00.00 migration/1
     7 root        15  -5     0     0     0  S   0  0.0   0:00.00 ksoftirqd/1
     8 root        RT  -5     0     0     0  S   0  0.0   0:00.00 watchdog/1
     9 root        RT  -5     0     0     0  S   0  0.0   0:00.02 migration/2
    10 root        15  -5     0     0     0  S   0  0.0   0:00.00 ksoftirqd/2
    11 root        RT  -5     0     0     0  S   0  0.0   0:00.00 watchdog/2
    12 root        RT  -5     0     0     0  S   0  0.0   0:00.01 migration/3
    13 root        15  -5     0     0     0  S   0  0.0   0:00.00 ksoftirqd/3
    14 root        RT  -5     0     0     0  S   0  0.0   0:00.00 watchdog/3
    15 root        15  -5     0     0     0  S   0  0.0   0:00.02 events/0
    16 root        15  -5     0     0     0  S   0  0.0   0:00.00 events/1

```

Figura 3.37 Uso de CPU cuando 4 MVs m1.small se están ejecutando y actualizan sus repositorios

Para este caso y de forma adicional, se ha utilizado el comando *mpstat*⁸⁰, que permite monitorear la actividad de los procesadores, de esta forma se puede observar en la Figura 3.38 que los valores de uso de CPU son similares a los de la Figura 3.37, por lo que podemos confirmar con una segunda herramienta de monitoreo, que al ejecutar un tipo diferente de MVs en las que se realizan acciones para sobrecargar el CPU, el porcentaje de uso de éste es menor que cuando se ejecuta un tipo más grande de máquina virtual.

```

root@nodo2: ~
root@nodo2:~# mpstat
Linux 2.6.31-22-server (nodo2) 01/04/2011 _x86_64_ (4 CPU)

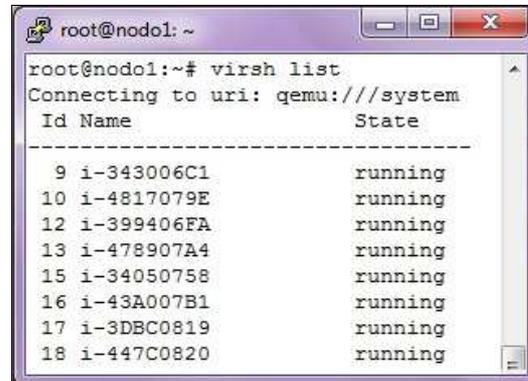
08:15:37 PM CPU %usr %nice %sys %iowait %irq %soft %steal %guest %idle
08:15:37 PM all 29.21 0.00 0.43 0.52 63.00 0.01 0.00 0.20 78.62

```

Figura 3.38 Visualización con *mpstat* del uso de CPU cuando 4 MVs m1.small actualizan sus repositorios

⁸⁰ *Mpstat* es una herramienta de línea de comandos para Linux que sirve para reportar estadísticas relacionadas al uso de CPU.

- c) El usuario Administrador y el usuario afrodita ejecutan en total 8 máquinas virtuales en el nodo 1, cada una de tipo c1.medium pero manejando el sistema operativo Ubuntu y Centos respectivamente; dichas MVs están listadas en la Figura 3.39.



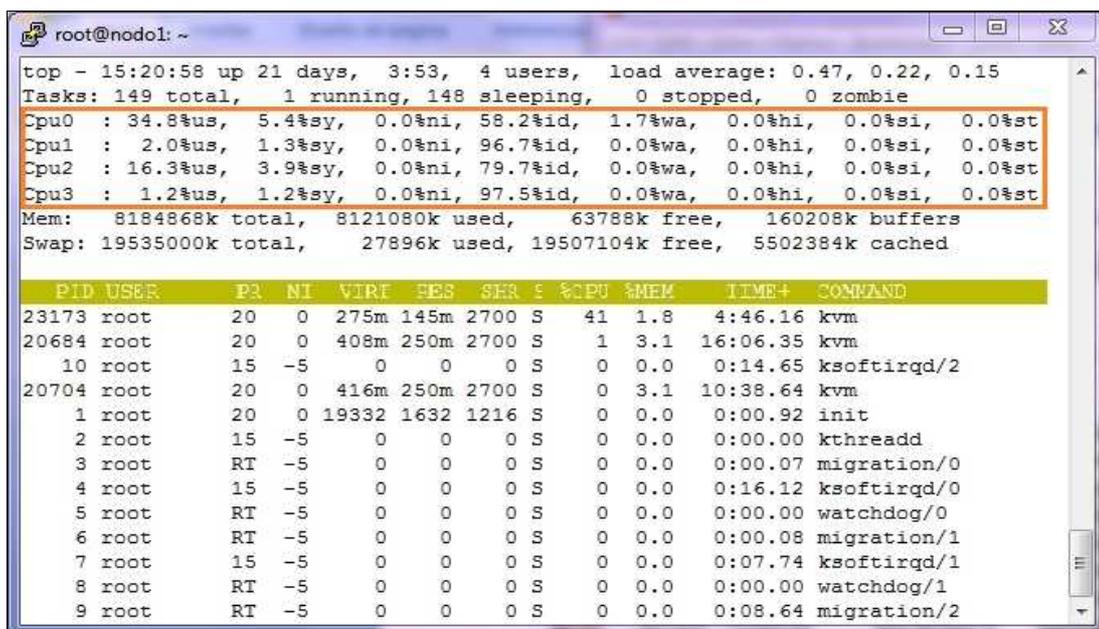
```

root@nodol:~# virsh list
Connecting to uri: qemu:///system
 Id Name                               State
-----
  9 i-343006C1                          running
 10 i-4817079E                          running
 12 i-399406FA                          running
 13 i-478907A4                          running
 15 i-34050758                          running
 16 i-43A007B1                          running
 17 i-3DBC0819                          running
 18 i-447C0820                          running

```

Figura 3.39 Listado de MVs ejecutándose en el nodo 1

En la Figura 3.40 al poner en marcha las 8 MVs en un solo nodo, el “%id”, tiempo en el que cada CPU no ha realizado ningún trabajo, ni tampoco ha esperado ninguna operación I/O, es del 58.2%, 96.7%, 79.7% y 97.5%, ya que el procesamiento en los CPUs aumenta en comparación al caso a), en donde en un solo nodo se ejecutan 4 MVs del mismo tipo.



```

top - 15:20:58 up 21 days, 3:53, 4 users, load average: 0.47, 0.22, 0.15
Tasks: 149 total, 1 running, 148 sleeping, 0 stopped, 0 zombie
Cpu0  : 34.8%us, 5.4%sy, 0.0%ni, 58.2%id, 1.7%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu1  :  2.0%us, 1.3%sy, 0.0%ni, 96.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu2  : 16.3%us, 3.9%sy, 0.0%ni, 79.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu3  :  1.2%us, 1.2%sy, 0.0%ni, 97.5%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem:   8184868k total, 8121080k used,    63788k free, 160208k buffers
Swap: 19535000k total,  27896k used, 19507104k free, 5502384k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 23173 root        20   0 275m 145m 2700 S   41   1.8   4:46.16 kvm
 20684 root        20   0 408m 250m 2700 S    1   3.1  16:06.35 kvm
    10 root        15  -5     0     0     0 S    0   0.0   0:14.65 ksoftirqd/2
 20704 root        20   0 416m 250m 2700 S    0   3.1  10:38.64 kvm
     1 root        20   0 19332 1632 1216 S    0   0.0   0:00.92 init
     2 root        15  -5     0     0     0 S    0   0.0   0:00.00 kthreadd
     3 root        RT  -5     0     0     0 S    0   0.0   0:00.07 migration/0
     4 root        15  -5     0     0     0 S    0   0.0   0:16.12 ksoftirqd/0
     5 root        RT  -5     0     0     0 S    0   0.0   0:00.00 watchdog/0
     6 root        RT  -5     0     0     0 S    0   0.0   0:00.08 migration/1
     7 root        15  -5     0     0     0 S    0   0.0   0:07.74 ksoftirqd/1
     8 root        RT  -5     0     0     0 S    0   0.0   0:00.00 watchdog/1
     9 root        RT  -5     0     0     0 S    0   0.0   0:08.64 migration/2

```

Figura 3.40 Uso de CPU cuando 8 MVs c1.medium arrancan en el nodo 1

Al momento de aumentar aún más el procesamiento de las CPUs (Figura 3.41), al actualizar los repositorios de las 8 MVs al mismo tiempo, se puede observar que el “%us” es del 10.5%, 11.6%, 11.3% y 13.3%, lo cual no representa un uso relativamente alto, pero el “%wa”, tiempo en el que cada CPU ha estado esperando ejecutar una operación es del 35.7%, 37.0%, 38.0% y 45.2% es mayor en comparación al de la Figura 3.40, esto es debido a que cada MV una vez que se ha puesto en marcha ya está realizando un procesamiento individual y adicional que afecte el uso de CPU.

```

root@nodol: ~
top - 15:43:34 up 21 days, 4:16, 3 users, load average: 0.52, 0.28, 0.19
Tasks: 147 total, 1 running, 146 sleeping, 0 stopped, 0 zombie
Cpu0  : 10.5%us,  3.0%sy,  0.0%ni, 50.5%id, 35.7%wa,  0.0%hi,  0.3%si,  0.0%st
Cpu1   : 11.6%us,  3.3%sy,  0.0%ni, 58.1%id, 37.0%wa,  0.0%hi,  0.5%si,  0.0%st
Cpu2   : 11.3%us,  3.0%sy,  0.0%ni, 58.7%id, 38.0%wa,  0.0%hi,  0.5%si,  0.0%st
Cpu3   : 13.3%us,  2.9%sy,  0.0%ni, 57.5%id, 45.2%wa,  0.0%hi,  0.7%si,  0.0%st
Mem:   8184868k total,  8134340k used,    50528k free,  160208k buffers
Swap: 19535000k total,  28544k used, 19506456k free,  5516900k cached

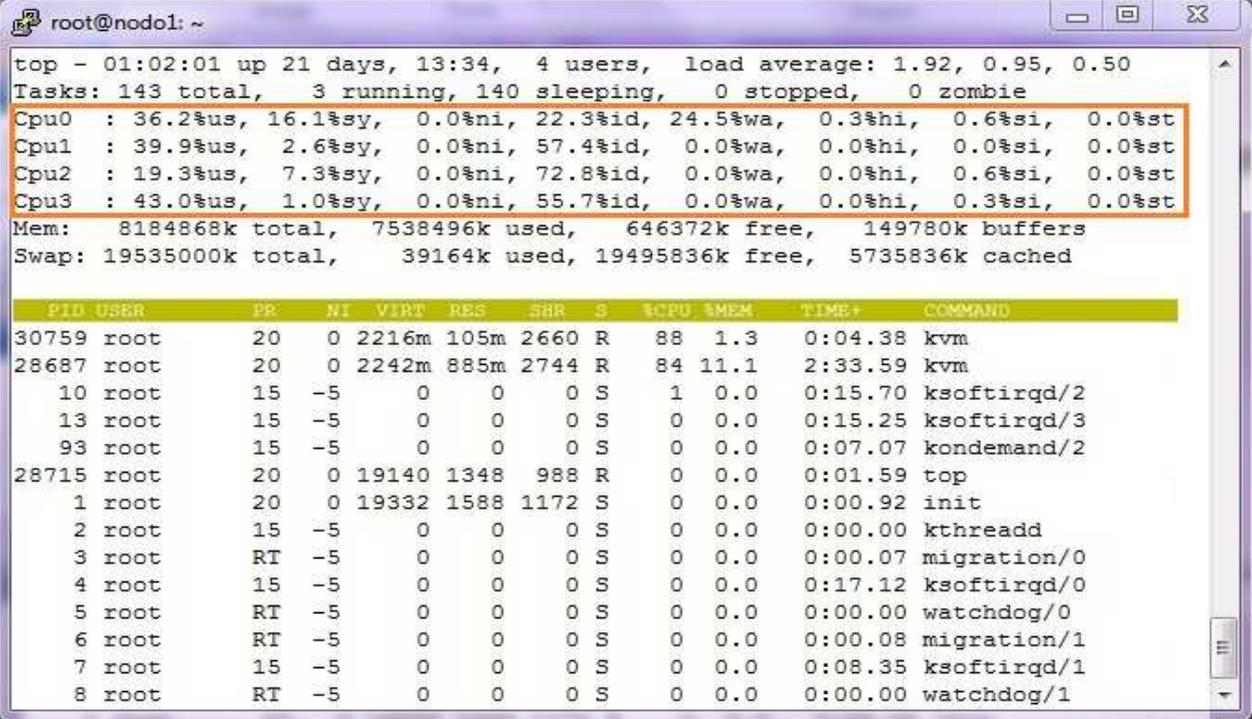
  PID USER      PR  NI  VIRT  RES  SHR  S   %CPU  %MEM    TIME+  COMMAND
 20698 root        20   0  424m 250m 2700 S    15   3.1   11:22.12 kvm
 16543 eucalypt    20   0  556m 254m 5412 S    1   3.2   19:25.46 apache2
 20684 root        20   0  408m 250m 2700 S    1   3.1   16:08.20 kvm
  1094 root        20   0 22188 1160 1044 S    0   0.0    3:27.40 hald-addon-stor
  1581 root        20   0 36644 1600  920 S    0   0.0   19:14.49 powernapd
    1 root        20   0 19332 1632 1216 S    0   0.0    0:00.92 init
    2 root        15  -5     0     0     0 S    0   0.0    0:00.00 kthreadd
    3 root         RT  -5     0     0     0 S    0   0.0    0:00.07 migration/0
    4 root        15  -5     0     0     0 S    0   0.0    0:16.15 ksoftirqd/0
    5 root         RT  -5     0     0     0 S    0   0.0    0:00.00 watchdog/0
    6 root         RT  -5     0     0     0 S    0   0.0    0:00.08 migration/1
    7 root        15  -5     0     0     0 S    0   0.0    0:07.78 ksoftirqd/1
    8 root         RT  -5     0     0     0 S    0   0.0    0:00.00 watchdog/1_

```

Figura 3.41 Uso de CPU cuando 8 MVs c1.medium se están ejecutando y actualizan sus repositorios

- d) En el nodo 1 es posible ejecutar un total de 2 MVs del tipo c1.xlarge, cada una de 4 CPUs, 2048 MB de memoria RAM y 20 GB en disco, sin embargo en la Figura 3.42 se puede visualizar que el uso de CPU a nivel de usuario cuando se ejecuta una sola MV es de 36.2%, 39.9%, 19.3% y 43.0%, mientras que el uso del CPU a nivel del sistema (*kernel*) es de 16.1%, 2.6%, 7.3% y 1.0%, estos últimos valores son superiores en comparación a los casos anteriores ya

que al ejecutar una MV de este tipo existe una mayor demanda del uso de CPU.



```

root@nodol: ~
top - 01:02:01 up 21 days, 13:34,  4 users,  load average: 1.92, 0.95, 0.50
Tasks: 143 total,  3 running, 140 sleeping,  0 stopped,  0 zombie
Cpu0  : 36.2%us, 16.1%sy,  0.0%ni, 22.3%id, 24.5%wa,  0.3%hi,  0.6%si,  0.0%st
Cpu1  : 39.9%us,  2.6%sy,  0.0%ni, 57.4%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu2  : 19.3%us,  7.3%sy,  0.0%ni, 72.8%id,  0.0%wa,  0.0%hi,  0.6%si,  0.0%st
Cpu3  : 43.0%us,  1.0%sy,  0.0%ni, 55.7%id,  0.0%wa,  0.0%hi,  0.3%si,  0.0%st
Mem:   8184868k total,  7538496k used,  646372k free,  149780k buffers
Swap: 19535000k total,   39164k used, 19495836k free,  5735836k cached

  PID USER      PR    NI  VIRT  RES  SHR  S   %CPU  %MEM    TIME+  COMMAND
 30759 root      20     0 2216m 105m 2660  R   88   1.3   0:04.38  kvm
28687 root      20     0 2242m 885m 2744  R   84  11.1   2:33.59  kvm
   10 root      15    -5     0     0     0  S    1   0.0   0:15.70  ksoftirqd/2
   13 root      15    -5     0     0     0  S    0   0.0   0:15.25  ksoftirqd/3
   93 root      15    -5     0     0     0  S    0   0.0   0:07.07  kondemand/2
28715 root      20     0 19140 1348  988  R    0   0.0   0:01.59  top
    1 root      20     0 19332 1588 1172  S    0   0.0   0:00.92  init
    2 root      15    -5     0     0     0  S    0   0.0   0:00.00  kthread
    3 root      RT    -5     0     0     0  S    0   0.0   0:00.07  migration/0
    4 root      15    -5     0     0     0  S    0   0.0   0:17.12  ksoftirqd/0
    5 root      RT    -5     0     0     0  S    0   0.0   0:00.00  watchdog/0
    6 root      RT    -5     0     0     0  S    0   0.0   0:00.08  migration/1
    7 root      15    -5     0     0     0  S    0   0.0   0:08.35  ksoftirqd/1
    8 root      RT    -5     0     0     0  S    0   0.0   0:00.00  watchdog/1

```

Figura 3.42 Uso de CPU cuando una MV c1.xlarge se ejecuta en el nodo1

Por otro lado, en la Figura 3.43, se muestran los resultados de uso de CPU cuando se ejecuta otra MV del mismo tipo, c1.xlarge, y así llegar a ejecutar el número máximo de MVs posibles en el nodo 1 (esto se puede saber utilizando el comando `euca-describe-availability-zones verbose` en el CC solo para el nodo1, cuando el servicio NC esté ejecutándose sólo en el nodo 1). Evidentemente en este caso el uso de CPU a nivel de usuario es casi el doble.

```

root@nodol: ~
top - 00:45:47 up 21 days, 13:18,  3 users,  load average: 1.36, 0.40, 0.19
Tasks: 140 total,  3 running, 137 sleeping,  0 stopped,  0 zombie
Cpu0  :  0.7%us,  0.0%sy,  0.0%ni, 99.3%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu1  : 83.2%us, 14.9%sy,  0.0%ni,  2.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu2  : 84.7%us, 14.0%sy,  0.0%ni,  1.3%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu3  : 83.1%us, 16.9%sy,  0.0%ni,  0.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   8184868k total,  5330304k used,  2854564k free,  149632k buffers
Swap: 19535000k total,   39168k used, 19495832k free,  4464108k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  MEM%   TIME+  COMMAND
 28687 root        20   0 2241m 119m 2724 S   297   1.5   0:34.84 kvm
16543 eucalypt    20   0  568m 264m 5412 S    1   3.3  21:21.55 apache2
   1 root         20   0 19332 1588 1172 S    0   0.0   0:00.92 init
   2 root         15  -5     0     0     0 S    0   0.0   0:00.00 kthreadd
   3 root         RT  -5     0     0     0 S    0   0.0   0:00.07 migration/0
   4 root         15  -5     0     0     0 S    0   0.0   0:17.06 ksoftirqd/0
   5 root         RT  -5     0     0     0 S    0   0.0   0:00.00 watchdog/0
   6 root         RT  -5     0     0     0 S    0   0.0   0:00.08 migration/1
   7 root         15  -5     0     0     0 S    0   0.0   0:08.28 ksoftirqd/1
   8 root         RT  -5     0     0     0 S    0   0.0   0:00.00 watchdog/1
   9 root         RT  -5     0     0     0 S    0   0.0   0:08.82 migration/2
  10 root         15  -5     0     0     0 S    0   0.0   0:15.56 ksoftirqd/2
  11 root         RT  -5     0     0     0 S    0   0.0   0:00.00 watchdog/2
  12 root         RT  -5     0     0     0 S    0   0.0   0:07.06 migration/3

```

Figura 3.43 Uso de CPU cuando una segunda MV c1.xlarge se está ejecutando en el nodo 1

- e) En el nodo 2 se ha ejecutado la única MV del tipo c1.xlarge que es posible arrancar. En la Figura 3.44 se puede observar que al ejecutar esta MV, el porcentaje de CPU que requiere el usuario “%us” no es sumamente alto, por lo que el “%id” de tiempo en que la CPU no trabaja ni ejecuta ninguna operación es mayor.

```

root@nodo2: ~
top - 00:43:22 up 4 days, 6:08, 2 users, load average: 0.86, 0.30, 0.16
Tasks: 137 total, 1 running, 136 sleeping, 0 stopped, 0 zombie
Cpu0  : 11.0%us, 11.3%sy, 0.0%ni, 76.7%id, 1.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu1  : 21.7%us, 9.9%sy, 0.0%ni, 68.4%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu2  : 16.2%us, 9.2%sy, 0.0%ni, 74.5%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu3  : 12.7%us, 12.1%sy, 0.0%ni, 75.2%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem:   8184868k total, 5341812k used, 2843056k free, 153332k buffers
Swap: 39070032k total, 0k used, 39070032k free, 4449332k cached

  PID USER      PR  NI  VIRT  RES  SHR  S   %CPU  %MEM    TIME+  COMMAND
 6810 root        20   0 2255m 130m 2732  S   109   1.6   0:29.47  kvm
    1 root         0   0 19332  1720 1216  S    0   0.0   0:00.92  init
    2 root        15  -5     0     0     0  S    0   0.0   0:00.00  kthreadd
    3 root        RT  -5     0     0     0  S    0   0.0   0:00.07  migration/0
    4 root        15  -5     0     0     0  S    0   0.0   0:00.44  ksoftirqd/0
    5 root        RT  -5     0     0     0  S    0   0.0   0:00.00  watchdog/0
    6 root        RT  -5     0     0     0  S    0   0.0   0:00.03  migration/1
    7 root        15  -5     0     0     0  S    0   0.0   0:00.12  ksoftirqd/1
    8 root        RT  -5     0     0     0  S    0   0.0   0:00.00  watchdog/1
    9 root        RT  -5     0     0     0  S    0   0.0   0:02.44  migration/2
   10 root        15  -5     0     0     0  S    0   0.0   0:00.12  ksoftirqd/2
   11 root        RT  -5     0     0     0  S    0   0.0   0:00.00  watchdog/2
   12 root        RT  -5     0     0     0  S    0   0.0   0:01.50  migration/3
   13 root        15  -5     0     0     0  S    0   0.0   0:00.13  ksoftirqd/3

```

Figura 3.44 Uso de CPU cuando 1 MV c1.xlarge se ejecuta en el nodo 2

Con todos los resultados anteriores, se puede concluir que al momento de ejecutar MVs en los nodos, éstas utilizan parte de todos los CPUs que posee cada nodo.

Al ejecutar en los dos nodos un total de 12 MVs del tipo m1.small, se está utilizando el mismo número de CPUs que al ejecutar en los dos nodos un total de 12 MVs del tipo c1.medium o 3 MVs del tipo c1.xlarge, ya que las MVs del tipo m1.small utilizan 1 CPU para cada MV, es decir 12 CPUs en total, mientras que las MVs del tipo c1.small también utilizan 1 CPU para cada MV, es decir 12 CPUs en total. Las MVs del tipo c1.xlarge utilizan 4 CPUs por cada MV, es decir 12 CPUs en total.

f) Finalmente, se ha decidido medir el uso de CPU cuando el nodo 1 puede ejecutar hasta un máximo de 16 MVs y el nodo 2 puede ejecutar hasta un máximo de 4 MVs de tipo m1.small o c1.medium (Sección 3.5).

```

root@nodo1: ~
top - 23:22:38 up 3 days, 2:31, 2 users, load average: 4.34, 1.19, 0.45
Tasks: 147 total, 9 running, 138 sleeping, 0 stopped, 0 zombie
Cpu0  : 32.9%us, 13.5%sy, 0.0%ni, 20.6%id, 31.6%wa, 0.0%hi, 1.3%si, 0.0%st
Cpu1  : 38.9%us, 18.6%sy, 0.0%ni, 26.4%id, 16.1%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu2  : 46.2%us, 48.5%sy, 0.0%ni, 4.9%id, 0.3%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu3  : 41.6%us, 13.5%sy, 0.0%ni, 35.5%id, 9.0%wa, 0.0%hi, 0.3%si, 0.0%st
Mem:   8184868k total, 8138192k used, 46676k free, 135660k buffers
Swap: 19535000k total, 33740k used, 19501260k free, 6446284k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 3930 root        20   0  429m  66m  2688  R   59   0.8   0:05.11  kvm
 3944 root        20   0  410m  64m  2688  R   52   0.8   0:04.41  kvm
 3922 root        20   0  409m  65m  2688  S   48   0.8   0:04.36  kvm
 3982 eucalypt   20   0 12388  724  564  R   39   0.0   0:01.17  cp
 3938 root        20   0  429m  59m  2688  S   26   0.7   0:03.49  kvm
 3952 root        20   0  409m  54m  2688  R   24   0.7   0:03.60  kvm
  47  root        15  -5    0    0    0  R    5   0.0   0:03.26  kswapd0
 3871 root        20   0  410m  83m  2696  S    2   1.0   0:05.41  kvm
1578 eucalypt   20   0  817m  214m 4860  S    1   2.7  16:24.82  apache2
2916 root        20   0  293m  147m 2712  S    1   1.8  10:52.07  kvm
3632 root        20   0  419m  94m  2700  S    1   1.2   0:08.93  kvm
  13  root        15  -5    0    0    0  S    0   0.0   0:03.14  ksoftirqd/3
  34  root        15  -5    0    0    0  S    0   0.0   0:04.32  ata/0
1141 root        20   0  336m  118m 2896  S    0   1.5   2:26.23  libvirtd

```

Figura 3.45 Uso de CPU cuando 16 MVs c1.medium se ejecutan en el nodo 1

```

root@nodo2: ~
top - 08:36:25 up 3 days, 11:46, 2 users, load average: 3.71, 2.25, 1.83
Tasks: 149 total, 2 running, 147 sleeping, 0 stopped, 0 zombie
Cpu0  : 11.0%us, 89.0%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu1  : 39.0%us, 13.7%sy, 0.0%ni, 47.3%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu2  : 35.5%us, 11.1%sy, 0.0%ni, 53.4%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Cpu3  : 23.3%us, 7.3%sy, 0.0%ni, 63.5%id, 6.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem:   8184868k total, 6755084k used, 1429784k free, 131064k buffers
Swap: 39070032k total, 88k used, 39069944k free, 5682308k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
10782 root        20   0  71916  30m  17m  R  100   0.4  4431:14  unattended-upgr
2032 root        20   0  389m  51m  2676  S   60   0.6   0:02.64  kvm
2026 root        20   0  389m  52m  2676  S   48   0.7   0:02.72  kvm
1923 root        20   0  420m  85m  2696  S   21   1.1   0:05.60  kvm
1857 root        20   0  409m  82m  2696  S    2   1.0   0:05.59  kvm
1127 root        20   0  265m  66m  3176  S    0   0.8   0:40.58  libvirtd
1564 eucalypt   20   0  788m  204m 5416  S    0   2.6  16:34.02  apache2
  1  root        20   0 19328  1688  1216  S    0   0.0   0:00.88  init
  2  root        15  -5    0    0    0  S    0   0.0   0:00.00  kthreadd
  3  root        RT  -5    0    0    0  S    0   0.0   0:00.76  migration/0
  4  root        15  -5    0    0    0  S    0   0.0   0:53.70  ksoftirqd/0
  5  root        RT  -5    0    0    0  S    0   0.0   0:00.00  watchdog/0
  6  root        RT  -5    0    0    0  S    0   0.0   0:00.01  migration/1
  7  root        15  -5    0    0    0  S    0   0.0   0:00.03  ksoftirqd/1

```

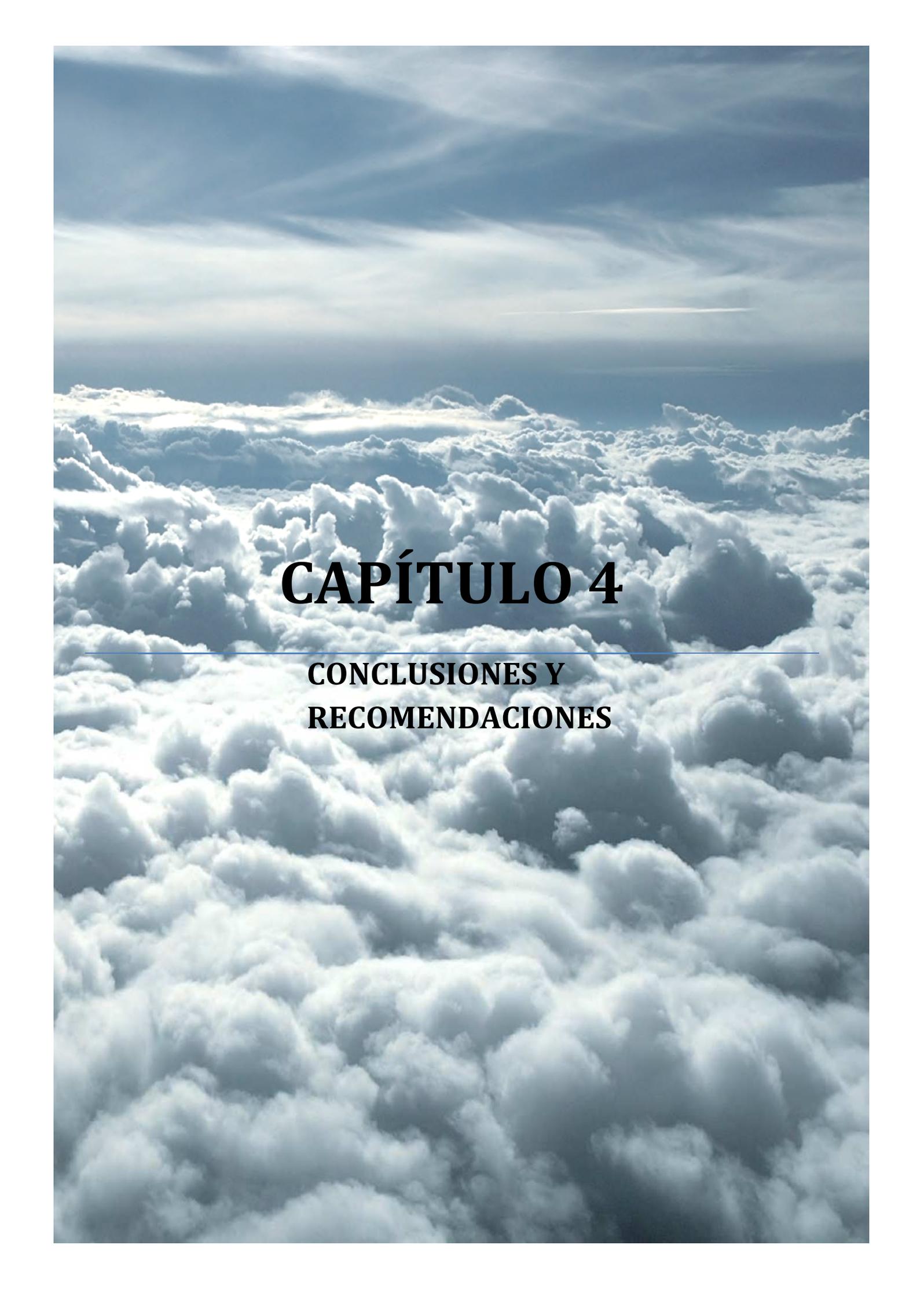
Figura 3.46 Uso de CPU cuando 4 MVs c1.medium se ejecutan en el nodo 2

A pesar de que en el nodo 1 se están ejecutando 16 MVs, en la Figura 3.45 se puede observar que el “%us”, uso de CPU a nivel de usuario en el nodo 1 no es alto, en comparación al que se puede visualizar en la Figura 3.46.

La ejecución simultánea de 16 MVs demanda de mayor tiempo que al ejecutar simultáneamente 4 MVs, por lo que el “%id”, tiempo en que la CPU ha estado sin trabajar y sin esperar operaciones para ejecutarlas, es menor en el nodo 1.

La Figura 3.46 permite visualizar que el “%us” es menor al 40%, mientras que los resultados del uso de CPU por el sistema (*kernel*) son similares a los de la Figura 3.45, ya que al obtener un promedio de estos valores; en el nodo 1 se conoce que el sistema utiliza un 23.53% mientras que en el nodo 2 se utiliza un 30.28%. El sistema en el nodo 2 podría utilizar mayor procesamiento en los 4 CPUs que forman parte de él, porque se puede aumentar el número de máquinas virtuales que podrían ejecutarse tal y como se hizo en el nodo 1.

En base a los resultados obtenidos se puede concluir que a pesar de estar ejecutando 4 veces el número de MVs que en el nodo 2, el uso de CPU en el nodo 1 no aumenta significativamente. De esta forma se puede recomendar que la configuración del número de máquinas virtuales en los nodos de las características especificadas en este prototipo, llegue hasta 16 MVs (cores) por cada uno (Sección 2.5.4.2); de tal forma que cuando se utilice al máximo cada máquina virtual, el uso total de la CPU sistema (*host*, Revisar Anexo B) no se vea altamente afectado. Con esto se asegura que la calidad del servicio se mantenga en un nivel adecuado, aprovechando al máximo los recursos de hardware disponibles.



CAPÍTULO 4

**CONCLUSIONES Y
RECOMENDACIONES**

4.1 CONCLUSIONES

- Comparando la instalación de *Eucalyptus* utilizando paquetes, con *Ubuntu Enterprise Cloud* (UEC), se concluye que es mucho más rápida y sencilla la alternativa con UEC, por lo que se reporta en este documento que la implementación del prototipo utilizó UEC que proporciona a los usuarios un conjunto de imágenes disponibles en el sitio oficial de *Ubuntu*, además de la facilidad que presenta para la implementación de Infraestructura como Servicio ya que los componentes de *Eucalyptus* se instalan junto con el sistema operativo dependiendo de la configuración que se utilice (nodo o *front-end*) además de elementos extra como el servidor de correo electrónico *Postfix*, el servidor web apache y el *hypervisor* KVM.
- Se presentaron diferentes alternativas de configuración de *Eucalyptus* en máquinas físicas. La implementación del prototipo permitió concluir que cada configuración depende de los requerimientos que se tengan, uno de los más importantes es mejorar el rendimiento general del sistema mediante la distribución de la carga de trabajo (Sección 2.2.1), por lo que se escogió la configuración *front-end* con los componentes *Walrus*, *Cloud*, *Cluster* y *Storage Controller* y dos nodos con el componente *Node Controller*.
- La instalación de los componentes de *Eucalyptus* distribuyéndolos en diferentes equipos, ha permitido comprobar que gracias a su modularidad, se puede escalar de una manera sencilla la capacidad de una Nube. Actualmente, el prototipo se compone de dos nodos controlados por un *Cluster Controller*; si la capacidad requerida del número de máquinas virtuales a ejecutarse se incrementa posteriormente, se puede optar por agregar más nodos, y si en un determinado momento la capacidad del *Cluster Controller* no es suficiente se puede agregar otro al *Cloud Controller*.

- Un usuario de una computadora trabajando en la Nube, deja de ser dependiente de una plataforma en particular (hardware más sistema operativo). En este proyecto se instalaron imágenes de dos sistemas operativos Linux (*Ubuntu* y *Centos*), que permiten ejecutar máquinas virtuales de ambos. La virtualización ha permitido que se ejecuten sistemas operativos *guest* diferentes del *host*; de esta forma se puede contar con distintas plataformas sin tener todas las aplicaciones en una sola.
- En el modo *Static* de *Eucalyptus* no se tiene aislamiento de tráfico entre máquinas virtuales ya que el usuario *root* puede interferir en el tráfico de red de cualquier máquina virtual. Por otro lado, en los modos *Managed* y *Managed No VLAN* se maneja mayor seguridad que en los modos *Static* y *System*, ya que se define una serie de grupos de seguridad a los que se pueden aplicar las reglas de entrada de tráfico; en el prototipo se pudo demostrar la flexibilidad en cuanto al control de red que se puede establecer en un sistema de Nube.
- Este proyecto ha permitido evidenciar la característica de control de usuarios de un sistema basado en Nube. Un usuario normal, es decir que no es administrador, no tiene control total sobre las funcionalidades del prototipo; por ejemplo, las imágenes en *Eucalyptus* pueden ser añadidas al sistema por cualquier usuario. Sin embargo, después que una imagen ha sido añadida, una instancia puede ser ejecutada únicamente por los usuarios a los cuales el administrador haya otorgado permisos; de igual manera la interfaz web presentada a un usuario normal no contiene las mismas opciones de configuración que un administrador.
- Con la implementación de este prototipo, se demostró que se puede instalar cualquier tipo de aplicación en una instancia, como se lo haría normalmente en una máquina, física o virtual, que no fuese parte de una Nube. Como ejemplo se utilizó el motor de base de datos *mysql* y el servidor web (*apache*) a través del cual se pudo publicar un sitio web; sin

embargo para mantener la persistencia de la información, los datos de las aplicaciones mencionadas se almacenaron en un volumen lógico.

- *Eucalyptus* permite utilizar cualquier procedimiento para mantener la persistencia de información. Ésta es la característica esencial del modelo de Infraestructura como Servicio puesto que el arranque de una máquina virtual se realiza en muy corto tiempo comparado con la instalación y configuración de una máquina física utilizando técnicas tradicionales. Es por tal razón que la persistencia de datos se realiza utilizando volúmenes lógicos en los cuales el método que se maneje para mantener la persistencia de los datos, depende de cada usuario.
- *Eucalyptus* permite el manejo de *snapshots*; con el prototipo implementado se pudo comprobar que los *snapshots* clonan a los volúmenes, incluyendo la información almacenada en ellos, lo cual representa una gran ventaja puesto que se puede multiplicar la información de los volúmenes en corto tiempo. A partir de un *snapshot* se pueden crear varios volúmenes que contienen la misma información del volumen clonado y además con distinta capacidad de almacenamiento, siempre y cuando no sea menor a la utilizada por la información almacenada en el *snapshot*.
- Las pruebas efectuadas en este proyecto son factibles de extensión a través del montaje o aprovechamiento de escenarios que involucren situaciones que puedan afectar el desempeño del modelo IaaS. El uso de CPU en los nodos no aumentará considerablemente cuando se ejecutan 12 máquinas virtuales de tamaño m1.small, ó 3 de tamaño c1.xlarge ya que se ha destinando la misma capacidad total de CPU en ambos casos para la ejecución de instancias simultáneamente. En la Figura 2.28 se presentó el máximo número y tipos de máquinas virtuales que pueden ejecutarse en el prototipo; para realizar una comparación en este caso se han considerado, 12 MVs pequeñas de 1 núcleo cada una ó 3 grandes de 4 núcleos cada una, en ambos casos el número total de núcleos es de 12. Sin embargo, se puede aumentar el número de máquinas virtuales que se pueden

ejecutar simultáneamente estableciendo la variable `MAX_CORES` de los nodos (Sección 2.5.4.2) en el valor requerido, pero en este caso el rendimiento de uso de CPU si se verá afectado⁸¹.

- Considerando el análisis de velocidad de transmisión y tráfico realizado con *iptraf*, *iftop* y MRTG, se ha concluido que el tráfico total generado por la interfaz de red externa (eth1) es inferior al tráfico total generado por la interfaz de red interna (eth0) ya que a través de la interfaz interna el *Cluster Controller* se comunica con los *Node Controllers* para realizar el cálculo de los recursos totales disponibles en el sistema y para realizar las peticiones de ejecución de máquinas virtuales, en este proceso se copian los archivos de la imagen del sistema operativo para arrancar cada instancia a través del *hypervisor*, por lo tanto la información enviada a través de esta interfaz de red es mayor en comparación con la interfaz externa.
- *Cloud Computing* es una tecnología que representa la evolución de la virtualización y los Servicios Web, esto ha permitido la interoperabilidad entre varias plataformas y la oferta de servicios bajo demanda. Dentro de la implementación de este proyecto se comprobó que se pueden ejecutar máquinas virtuales desde un equipo que tenga instalado el navegador web *Mozilla Firefox*, el cual es gratuito, pero también existen herramientas comerciales como *RightScale* (Revisar Anexo G) que permite administrar y gestionar una Nube de cualquier proveedor y desde cualquier plataforma con cualquier navegador web. Además de la interfaz web, también se puede usar la línea de comandos para la administración de la Nube (*euca2ools*), lo que demuestra la flexibilidad para el uso de diferentes herramientas que permitan la gestión de los servicios que ofrece la una Nube basada en *Eucalyptus*.

⁸¹ http://open.eucalyptus.com/wiki/EucalyptusTroubleshooting_v2.0#cores

4.2 RECOMENDACIONES

- Cuando un usuario utiliza la administración de zonas de disponibilidad, (*availability-zones*), y no puede observar el nombre del *cluster* en el que puede ejecutar una instancia, es porque el *Cloud Controller* no tiene comunicación con el *Cluster Controller*. Esto puede suceder porque no se ha registrado el *Cluster Controller* con el *Cloud Controller* o existe un conflicto con las credenciales de usuario. Adicionalmente, si la administración de disponibilidad de zonas, no muestra al administrador la cantidad de recursos disponibles para que cualquier usuario pueda ejecutar máquinas virtuales en un determinado nodo, es porque el *Node Controller* no puede comunicarse correctamente con el *Cluster Controller*, por ésto si se añaden nodos al prototipo es indispensable sincronizar las llaves de éstos.
- Para que un usuario pueda acceder a una máquina virtual en ejecución a través de un cliente SSH, lo primero que se debe hacer es autorizar el acceso al puerto 22 para que acepte dichas conexiones y se eviten errores de acceso. Esta acción se la realiza a través de los grupos de seguridad. Para acceder desde una red externa a la red privada de las máquinas virtuales es necesario que este puerto también esté abierto. Dentro del proyecto se utilizó la red de la EPN como una red “pública” y no se tuvo ningún inconveniente con la apertura del puerto ya que esta red tiene abierto este puerto por defecto.
- Cuando se quiere acceder a una máquina virtual a través de la línea de comandos, se puede presentar el error “WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!”, lo cual implica que se le ha asignado la misma dirección IP de una máquina virtual que ya fue terminada a una nueva, esto se debe a que existe una reutilización de direcciones IP. Para solucionar este inconveniente se debe borrar del archivo (`/root/.ssh/known_hosts:21`), la línea especificada en el error presentado,

en este caso, la línea 21 que es la que corresponde a la identificación fallida.

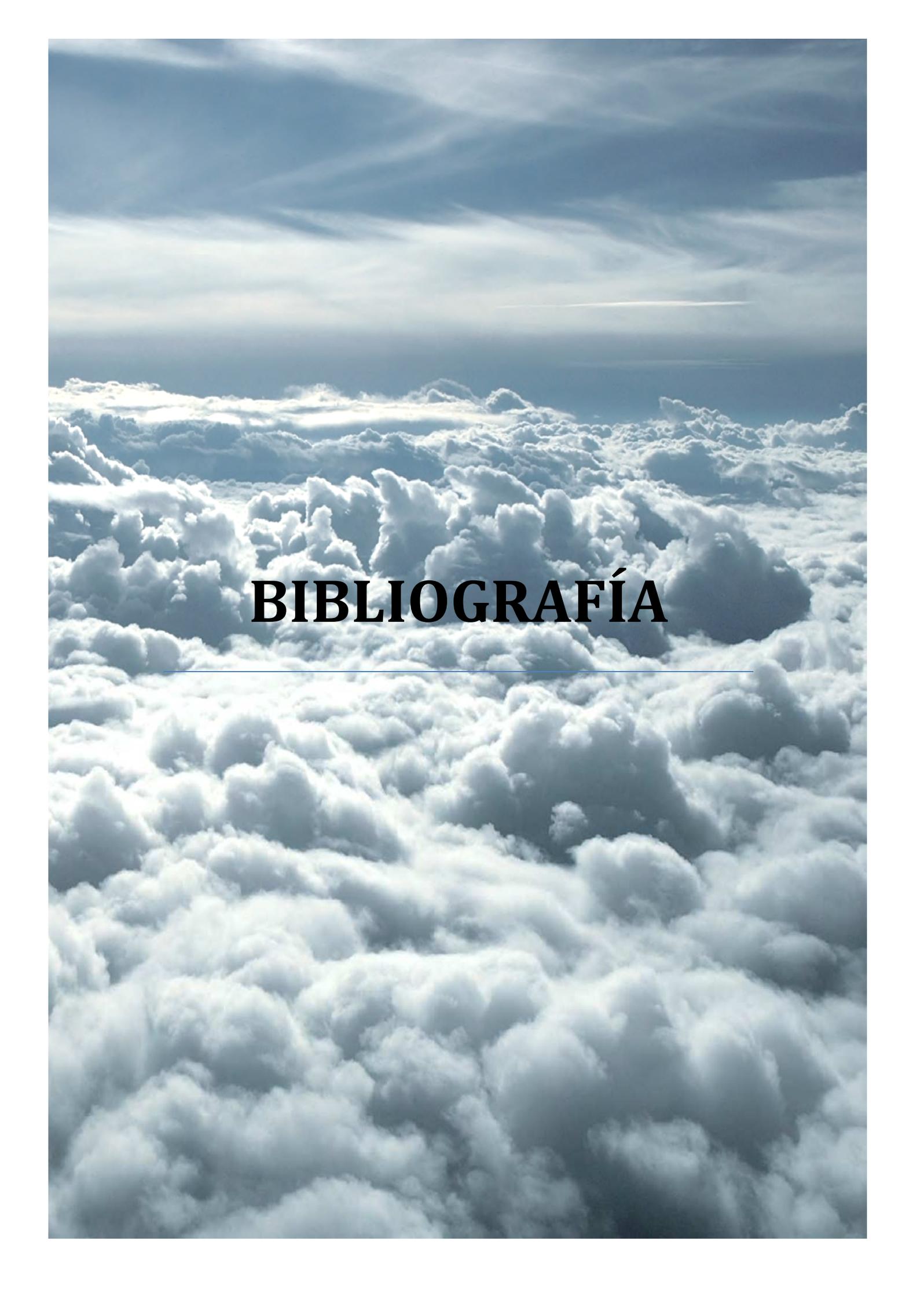
- Para utilizar una herramienta de gestión de Nube fue necesario abrir los puertos 8773 (*Walrus*) y 8443 (*Cloud Controller*) en el *firewall* de la red en la que se encuentra implementado el prototipo; esto para acceder a la Nube con *Elasticfox* y *Rightscale* desde el Internet, para lo cual es necesario realizar una solicitud dirigida a las personas encargadas de la administración de la red de la EPN para la apertura de los puertos requeridos.
- La llave privada para la conexión con una máquina virtual a través de *putty* tiene que ser convertida a la extensión que soporta *putty* (.ppk). Si se generan *keypairs* a través de la línea de comandos la extensión que se puede usar para almacenar la llave privada es (.priv ó .private) pero si se requiere usar esta misma llave con *putty*, se tendrá que usar *puttygen* para la conversión de dicha llave a (.ppk). Si se generan *keypairs* a través de *Elasticfox* la extensión por defecto para la llave privada es (.pem), la cual también deberá ser convertida a (.ppk) si se la quiere utilizar con *putty*, para esto también se puede usar *puttygen*.
- Si al tratar de ejecutar una máquina virtual el servidor DHCP no le asigna la dirección IP, permanecerá en estado de espera (*pending*) y jamás llegará al estado de ejecución (*running*), es por tal razón que se tiene que revisar el servicio del componente de *Eucalyptus* encargado de realizar esta acción, por lo tanto, es recomendable que se reinicie el *Cluster Controller* utilizando el comando (`restart eucalyptus-cc CLEAN =1`), lo cual no afectará de ninguna manera al funcionamiento normal del resto de componentes.
- Para solucionar posibles problemas que se presenten dentro del prototipo, ya sea comunicación entre componentes, acceso vía web o ejecución de máquinas virtuales, es recomendable revisar los registros (logs) de *Eucalyptus* y del sistema, tanto en el *front-end* como en los nodos, ya que

éstos poseen toda la información correspondiente al funcionamiento de la Nube; el directorio por defecto es `/var/log/eucalyptus` y el nivel de escritura de los registros en *Eucalyptus* se puede configurar en el archivo `eucalyptus.conf` con la variable `LOGLEVEL` (Sección 2.5.4.2).

- En el presente proyecto se ejecutaron máquinas virtuales que al visualizarlas en *Elasticfox* parecían haber finalizado de forma casi instantánea, mientras que al visualizarlas dentro de cada nodo estas máquinas aún permanecían en ejecución, este problema puede presentarse debido a que existe un error de comunicación de la herramienta *libvirt*, la cual permite el manejo de la plataforma de virtualización, con el *Node Controller*. Para solucionar este problema es necesario revisar los registros (logs) de *Eucalyptus* en los nodos y la ejecución de los servicios tanto del *Node Controller* (`eucalyptus-nc`) como de *libvirt* (`libvirt-bin`), si presentan algún inconveniente es necesario reiniciarlos.
- Se recomienda que los nodos que forman parte de la implementación de una Nube como la descrita en este proyecto, manejen hardware que soporte extensión de virtualización (Intel-VT o AMD-V), ya que se utilizó el *hypervisor* KVM que permite acelerar el proceso de ejecución de máquinas virtuales al ser un *hypervisor* que utiliza virtualización completa (Anexo B.2.3). Al inicio del proyecto, la instalación de los nodos se la realizó utilizando equipos que no soportaban extensión de virtualización y por lo tanto se utilizó otro *hypervisor* llamado QEMU que provocó que la ejecución de las máquinas virtuales demandase mayor tiempo.
- Para un manejo más sencillo de la Nube, lo más recomendable es utilizar la interfaz web a través de la herramienta de gestión *Elasticfox*. Cuando se presentó el plan del proyecto se especificó a *Elasticfox* como herramienta de gestión gráfica, sin embargo en el transcurso del desarrollo del proyecto se probó otra herramienta un poco más avanzada que *Elasticfox*, llamada *Hybridfox* la cual permite incluso ejecutar máquinas virtuales utilizando

direccionamiento privado, esto se requiere cuando no existen suficientes direcciones IP públicas para asignar a las máquinas virtuales. *Hybridfox* mantiene la misma interfaz gráfica de *Elasticfox* y al igual que éste es un *plugin* de *Mozilla Firefox*. Existen varias versiones de *Hybridfox*, la que se recomienda al momento es la 1.6.000019⁸².

⁸² Enlace de descarga: <http://code.google.com/p/hybridfox/downloads/detail?name=hybridfox-1.6.000019.xpi&can=2&q=>



BIBLIOGRAFÍA

- [1] “*Qué es Cloud Computing*”
 Autor: Salesforce Inc.
<http://www.salesforce.com/es/cloudcomputing>
 Última visita: julio 31, 2010.
- [2] LÁSTRAS, Jorge; LÁZARO, Javier; MIRÓN, Jonatan. “*Arquitecturas de red para servicios en Cloud Computing*”. España, 2007”. PDF
 Disponible en: <http://eprints.ucm.es/9452/>
 Última visita: julio 1, 2010.
- [3] “*Introduction to server virtualization*”
 Autor: OU, George.
<http://search.techrepublic.com.com/search/George%20Ou.html>
 Última visita: julio 20, 2010.
- [4] “*Comité de Tecnología de la Información: Cloud Computing*”
 Autor: MARTÍNEZ, Adriana.
http://www.venamcham.org/index.php?option=com_content&view=article&id=242%3Acomite-de-tecnologia-de-la-informacion-cloud-computing&catid=8%3Acomitesaldia&lang=es
 Última visita: septiembre 5, 2010.
- [5] “*SOAP*”
 Autor: Wikipedia.
<http://en.wikipedia.org/wiki/SOAP>
 Última visita: julio 25, 2010.
- [6] “*REST*”
 Autor: Wikipedia.
http://es.wikipedia.org/wiki/Representational_State_Transfer
 Última visita: julio 25, 2010.
- [7] “*Eucalyptus Cloud Computing Platform, User Guide v 1.6*”
 Autor: Eucalyptus Systems Inc.
 Disponible en: <http://open.eucalyptus.com/EucalyptusUserGuide.v1.final.03.23.pdf>
 Última visita: septiembre 28, 2010.
- [8] GEER, David. “*The OS Faces a Brave new world*”. Innovative Technology for Computer. IEEE Computer society. Octubre 2009.
 Disponible en:
http://books.google.com.ec/books?id=xfB5IH9cVSUC&pg=PA121&dq=%E2%80%9CThe+OS+Faces+a+Brave+new+world%22&hl=es&ei=E0T9TICbFIGB8ga6vsHjCg&sa=X&oi=book_result&ct=result&resnum=1&ved=0CCcQ6AEwAA#v=onepage&q&=false
 Última visita: septiembre 30, 2010.
- [9] “acens Technologies S.A. “*Información básica. ¿Qué es el SLA?*”. Madrid, 2010. PDF
 Disponible en:
[http://www.google.com.ec/url?sa=t&source=web&cd=4&ved=0CCsQFjAD&url=http%](http://www.google.com.ec/url?sa=t&source=web&cd=4&ved=0CCsQFjAD&url=http%3A)

[3A%2F%2Fwww.acens.com%2Ffile_download%2F176&ei=ZS87TNbbB8OAIafJ6_XVBw&usq=AFQjCNEm2x-WXF1nkjsSOFu_tyP5rZqVuw](http://www.acens.com/file_download/176&ei=ZS87TNbbB8OAIafJ6_XVBw&usq=AFQjCNEm2x-WXF1nkjsSOFu_tyP5rZqVuw)

Última visita: julio 15, 2010.

[10] *“What is a colocation facility?”*

Autor: KIETZMAN, Shannon.

<http://www.wisegeek.com/what-is-a-colocation-facility.htm>

Última visita: julio 2, 2010.

[11] *“Cloud Computing: nueva era de desarrollo”*

Autor: ROCHE, Stephanie.

<http://www.maestrosdelweb.com/editorial/cloud-computing-nueva-era-de-desarrollo/>

Última visita: julio 2, 2010.

[12] *“Outsourcing”*

Autor: ROMERO, Antonio.

<http://www.gestiopolis.com/recursos/documentos/fulldocs/ger/outsourcingantonio.htm>

Última visita: julio 5, 2010.

[13] Sun Microsystems, white paper, *“Introduction to Cloud Computing Architecture”*, 1ra edición, junio 2009.

Disponible en:

http://books.google.com.ec/books?id=jLNGCPs6rr4C&pg=PA254&dq=%E2%80%9CIntroduction+to+Cloud+Computing+Architecture%22&hl=es&ei=wkP9TPOMA4L58Aa2genoCg&sa=X&oi=book_result&ct=result&resnum=1&ved=0CCUQ6AEwAA#v=onepage&q=%E2%80%9CIntroduction%20to%20Cloud%20Computing%20Architecture%22&f=false

Última visita: septiembre 5, 2010.

[14] *“Hypervisor”*

Autor: Desconocido.

http://www.virtualizacion.com/?page_id=8

Última visita: julio 5, 2010.

[15] *“Virtualización de la Plataforma”*

Autor: Wikipedia.

http://es.wikipedia.org/wiki/Virtualizaci%C3%B3n#Virtualizaci.C3.B3n_de_plataforma

Última visita: julio 12, 2010.

[16] VELTE, Anthony T.; VELTE, Toby J.; ELSENPETER, Robert. *“Cloud Computing: A Practical Approach”*. The McGraw-Hill Companies, 2010.

Disponible en:

http://books.google.com.ec/books?id=mf0LMXve2gEC&printsec=frontcover&dq=++%E2%80%9CCloud+Computing:+A+Practical+Approach%E2%80%9D.&hl=es&ei=c0P9TOPeNsP88AahqNX7Cg&sa=X&oi=book_result&ct=result&resnum=1&ved=0CCUQ6AEwAA#v=onepage&q&f=false

Última visita: septiembre 12, 2010.

[17] *“IBM y Cloud Computing”*

Autor: globaltalentbox.wordpress.

<http://globaltalentbox.wordpress.com/2010/03/29/ibm-y-cloud-computing/>

Última visita: julio 8, 2010.

- [18] *“Cloud Myths Dispelled”*
 Autor: Eucalyptus Systems, Inc.
<http://www.eucalyptus.com/resources/info/cloud-myths-dispelled>
 Última visita: julio 11, 2010.
- [19] *“Virtualización y Servicios distribuidos en Red (Cloud Computing): impacto en los proveedores de servicio”*
 Autor: Desconocido.
<http://www.mkm-pi.com/mkmpi.php?article3924>
 Última visita: julio 18, 2010.
- [20] Cloud Security Alliance. *“Guía para la Seguridad en áreas críticas de atención en Cloud Computing”*. Noviembre 2009”. PDF
 Disponible en: <http://www.cloudsecurityalliance.org/guidance/csaguide-es.v2.pdf>
 Última visita: julio 20, 2010.
- [21] *“Mitos y Verdades del Cloud Computing”*
 Autor: CIO América Latina.
<http://www.cioal.com/analisis/arquitectura/mitos-y-verdades-del-cloud-computing.html>
 Última visita: julio 12, 2010.
- [22] *“Blog S21sec – Aclarando conceptos.SaaS, IaaS y PaaS”*
 Autor: SANESTEBAN, Juan.
<http://blog.s21sec.com/2009/07/aclarando-conceptos-saas-iaas-y-paas.html>
 Última visita: julio 22, 2010.
- [23] *Como el “Cloud Computing” está cambiando al Mundo!*
 Autor: REY, Rachael.
<http://bgnetworks.com.mx/blog/tag/saas/>
 Última visita: julio 15, 2010.
- [24] *“La gestión de Servicios en modo SaaS es lo que más está creciendo”*
 Autor: Desconocido.
<http://www.muycanal.com/2010/03/02/la-gestion-de-servicios-en-modo-saas-es-lo-que-mas-esta-creciendo>
 Última visita: septiembre 3, 2010.
- [25] MATHER, Tim; KUMARASWAMY, Subra; LATIF Shahed. *“Cloud Security and Privacy. An Enterprise Perspective on Risks and Compliance”*. O’Reilly.
 Disponible en:
<http://books.google.com.ec/books?hl=es&lr=&id=BHazecOuDLYC&oi=fnd&pg=PR11&dq=++%E2%80%9Ccloud+Security+and+Privacy.+An+Enterprise+Perspective+on+Risks+and+Compliance&ots=Fx86N4kTA9&sig=2OhtYG6WWGhKJFrMWRVrouiQV7Y#v=onepage&q&f=false>
 Última visita: septiembre 5, 2010.

- [26] *"Mashup"*
 Autor: Wikipedia.
[http://es.wikipedia.org/wiki/Mashup_\(aplicaci%C3%B3n_web_h%C3%ADbrida\)](http://es.wikipedia.org/wiki/Mashup_(aplicaci%C3%B3n_web_h%C3%ADbrida))
 Última visita: julio 15, 2010.
- [27] WEINHARDT, Christof; ANANDASIVAM, Arun; BLAU, Benjamin; STÖBER, Jochen. *"Business Models in the Service World"*. Technology for Computer, IEEE Computer society, marzo 2009.
 Disponible en:
http://pub.benjaminblau.de/Business_Models_in_the_Service_World.pdf
 Última visita: julio 18, 2010.
- [28] *"Cloud Computing, specialist virtualization kit"*.
 Autor: MENKEN, Ivanka.
<http://www.theartofservice.com/>
 Última visita: julio 15, 2010.
- [29] *"Qué es la Web 2.0"*
 Autor: Desconocido
https://www.microsoft.com/business/smb/eses/internet/web_2.msp
 Última visita: julio 11, 2010.
- [30] *"Web 3.0, la próxima generación de la red"*
 Autor: GUTIÉRREZ, Maite.
http://www.lavanguardia.es/premium/publica/publica?COMPID=53460692833&ID_PAGINA=22088&ID_FORMATO=9&turbourl=false
 Última visita: julio 7, 2010.
- [31] *"Evolución de la web"*
 Autor: GUERRA, Miguel; MIRANDA, Vinicio; RODAS, Carlos.
http://www.youtube.com/watch?v=MGnvPWyptjE&feature=player_embedded
 Última visita: julio 5, 2010.
- [32] *"Utility Computing"*
 Autor: Tech Target.
http://searchdatacenter.techtarget.com/sDefinition/0,,sid80_gci904539,00.html
 Última visita: julio 2, 2010.
- [33] *"Utility Computing"*
 Autor: Wikipedia.
http://es.wikipedia.org/wiki/Utility_computing
 Última visita: julio 13, 2010.
- [34] *"Above the Clouds: A Berkeley View of Cloud Computing"*
 Autor: AMBRUST, Michael; FOX, Armando; GRIFFITH, Rean; JOSEPH, Anthony; KATZ, Randy; KONWINSKI, Andrew; LEE, Gunho; PATTERSON, Davif; RABKIN, Ariel; STOICA, Ion y ZAHARIA, Matei.
<http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>
 Última visita: septiembre 15, 2010.

- [35] MILLER, Michael. *“Cloud computing, web based applications, that change The Way you work and Collaborate on Online”*. Safari books online, 2008.
- [36] REESE, George. *“Cloud Application Architecture. Building application and infrastructure in the Cloud”*. O’Reilly, Safari books online, 2009, EEUU, 2009.
- [37] *“Eucalyptus Systems”*
 Autor: Wikipedia.
http://en.wikipedia.org/wiki/Eucalyptus_%28computing%29
 Última visita: julio 21, 2010.
- [38] *“The Open Source Cloud Platform”*
 Autor: Eucalyptus Systems, Inc.
<http://open.eucalyptus.com/>
 Última visita: julio 21, 2010
- [39] NURMI, Daniel; WOLSKI, Rich; GRZEGORCZYK, Chris; OBERTELLI, Graziano; SOMAN, Sunil; YOUSEFF, Lamia; ZAGORODNOV, Dmitri. *“The Eucalyptus Open-source Cloud-computing System”*. Computer Science Department, University of California, Santa Barbara-California.
 Disponible en:
<http://portal.acm.org/purchase.cfm?id=1577895&CFID=550297&CFTOKEN=97498852>
 Última visita: julio 20, 2010.
- [40] *“EUCALYPTUS, computación en la nube con código abierto”*
 Autor: Desconocido.
<http://www.socialgnu.org/index.php/noticias/199-eucalyptus-computacion-en-la-nube-con-codigo-abierto>
 Última visita: julio 21, 2010.
- [41] *“Eucalyptus Open-Source Cloud Computing Infrastructure - An Overview”*
 Autor: Eucalyptus Systems, Inc.
<http://open.eucalyptus.com/search/node/%E2%80%9CEucalyptus%20Open-Source%20Cloud%20Computing%20Infrastructure%20-%20An%20Overview%E2%80%9D>
 Última visita: septiembre 21, 2010.
- [42] *“A Technical Report on an Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems”*
 Autor: NURMI, Daniel; WOLSKI, Rich; GRZEGORCZYK, Chris; OBERTELLI, Graziano; SOMAN, Sunil; YOUSEFF, Lamia y ZAGORODNOV Dmitrii.
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.154.2138>
 Última visita: octubre 18, 2010.
- [43] *“Eucalyptus”*
 Autor: Desconocido
<http://l10n.ubuntu.tla.ro/ubuntu-docs/jaunty/html/serverguide/es/eucalyptus.html>
 Última visita: junio 22, 2010.

- [44] JOHNSON, D; MURARI, Kiran; RAJU, Murthy; SUSEENDRAN, RB; YOGESH, Girikumar; *“Eucalyptus Beginner's Guide UEC Edition”* (Ubuntu Server 10.04 - Lucid Lynx), v1.0, 25 mayo 2010. PDF
 Disponible en:
http://cssoss.files.wordpress.com/2010/06/book_eucalyptus_beginners_guide_uec_edition1.pdf
 Última visita: julio 15, 2010.
- [45] *“Overview of Eucalyptus”*,
 Autor: RightScale, Inc.
http://support.rightscale.com/09-Clouds/Eucalyptus/01-Overview_of_Eucalyptus
 Última visita: julio 12, 2010.
- [46] *“PFC - Migración de un entorno web a Cloud Computing Amazon EC2”*. PDF
 Disponible en:
http://upcommons.upc.edu/pfc/bitstream/2099.1/8966/1/Memoria_PFC_Cloud_Computing.pdf
 Última visita: noviembre 12, 2010.
- [47] BONILLA, Jorge; CARRASCO, Daniel. Proyecto de Titulación: *“Análisis en implementación de un prototipo de servidor virtualizado sobre una distribución de Linux para el uso en Pymes”*. Escuela Politécnica Nacional, Facultad Ingeniería Eléctrica y Electrónica. Quito, febrero 2010.
- [48] *“Hypervisor”*
 Autor: Wikipedia.
<http://es.wikipedia.org/wiki/Hipervisor>
 Última visita: agosto 22, 2010.
- [49] *“¿Qué es un RAMDISK?”*
 Autor: JONES, Dilwyn.
<http://www.speccy.org/sinclairql/articulos/extensiones/ramdisk.htm>
 Última visita: agosto 18, 2010.
- [50] *“Ubuntu”*
 Autor: Desconocido.
<http://www.flickr.com/photos/23701795@N02/3933701075>
 Última visita: agosto 18, 2010.
- [51] *“Ubuntu_karmic_koala_9_10”*
 Autor: Desconocido.
http://planetared.com/wp-content/uploads/2009/10/Ubuntu_karmic_koala_9_10.jpg
 Última visita: octubre 7, 20010.
- [52] *Cloud Computing: Una buena forma de “estar en las nubes...”*
 Autor: PUEYO, Jorge.
<http://www.trume.com/2010/04/16/cloud-computing-una-buena-forma-de-%E2%80%9Cestar-en-las-nubes%E2%80%A6%E2%80%9D/>
 Última visita: agosto 26, 2010.

- [53] *"A Taxonomy and Survey of Cloud Computing Systems"*
Autor: IEEE.
www.ieee.org
Última visita: agosto 29, 2010.
- [54] *"Xen"*
Autor: Wikipedia.
<http://en.wikipedia.org/wiki/Xen>
Última visita: agosto 31, 2010.
- [55] *"KVM, ¿el nuevo rey de la virtualización?"*
Autor: Desconocido.
<http://www.muylinux.com/2009/10/18/kvm-%C2%BFel-nuevo-rey-de-la-virtualizacion>
Última visita: octubre 1, 2010.
- [56] *"Qemu"*
Autor: Wikipedia.
<http://en.wikipedia.org/wiki/Qemu>
Última visita: agosto 25, 2010.
- [57] *"BinaryTranslation"*
Autor: Wikipedia.
http://en.wikipedia.org/wiki/Binary_translation
Última visita: julio 30, 2010.
- [58] *"Qemu - Open Source Processor Emulator"*
Autor: Desconocido.
<http://wiki.qemu.org/Index.html>
Última visita: julio 31, 2010.
- [59] *"The GTK+ Project"*
Autor: GTK+ Team.
<http://www.gtk.org/>
Última visita: agosto 14, 2010.
- [60] *"Qemu"*
Autor: Desconocido.
<http://doc.ubuntu-es.org/index.php?title=QEmu>
Última visita: agosto 17, 2009.
- [61] *"Improving Security on EC2 with AWS Identity and Access Management (IAM)"*
Autor: HAMMOND, Eric.
<http://cloud.ubuntu.com/>
Última visita: agosto 21, 2010.
- [62] *"VMware"*
Autor: Wikipedia.
<http://es.wikipedia.org/wiki/VMware>

Última visita: julio 31, 2010.

- [63] *“Why Choose, VMware - Choose VMware for Greater Efficiency, Control and Flexibility”*

Autor: VMware, Inc.

<http://www.vmware.com/virtualization/why-choose-vmware.html>

Última visita: agosto 29, 2010.

- [64] *“Amazon Web Services”*

Autor: Wikipedia.

http://en.wikipedia.org/wiki/Amazon_AWS

Última visita: agosto 29, 2010.

- [65] *“¿Qué usos tiene Amazon Web Services?”*

Autor: Saasmania.

<http://www.saasmania.com/2010/01/05/%C2%BFque-usos-tiene-amazon-web-services/>

Última visita: septiembre 5, 2010.

- [66] *“Interacting with Block Storage (1.5.1)”*

Autor: Eucalyptus Systems, Inc.

http://open.eucalyptus.com/wiki/EucalyptusBlockStoreInteracting_v1.5

Última visita: agosto 18, 2010.

- [67] *“Interacting with Block Walrus (1.5.1)”*

Autor: Eucalyptus Systems, Inc.

http://open.eucalyptus.com/wiki/EucalyptusWalrusInteracting_v1.5

Última visita: septiembre 20, 2010.

- [68] *“Intel Virtualization Technology”*

Autor: Webopedia tm.

http://www.webopedia.com/TERM/I/Intel_Virtualization_Technology.html

Última visita: agosto 15, 2010.

- [69] Intel Corporation. *“Procesador de escritorio Intel Core i5”*

Disponible en: http://cache-www.intel.com/cd/00/00/43/84/438433_438433.pdf

Última visita: septiembre 20, 2010.

- [70] *“Virtualization”*

Autor: Intel Corporation.

<http://www.intel.com/technology/virtualization/>

Última visita: junio 14, 2010.

- [71] *“Intel Virtualization Technology (VT) Explained, Gabriel Torres”*

Autor: TORRES, Gabriel.

<http://www.hardwaresecrets.com/article/Intel-Virtualization-Technology-VT-Explained/263/2>

Última visita: octubre 16, 2010.

- [72] *“Tecnología AMD de Virtualización”*
Autor: Advanced Micro Devices, Inc.
<http://sites.amd.com/es/business/it-solutions/virtualization/Pages/amd-v.aspx>
Última visita: septiembre 18, 2010.
- [73] *“Características básicas de arquitectura de los procesadores AMD Phenom™ X3 Triple-Core, Anónimo”*
Autor: Advanced Micro Devices, Inc.
<http://www.amd.com/es/products/desktop/processors/phenom/Pages/AMD-phenom-processor-X3-features.aspx>
Última visita: julio 28, 2010.
- [74] *“Definición de GNU/ GRUB”*
Autor: ALEGSA.
<http://www.alegsa.com.ar/Dic/gnu%20grub.php>
Última visita: agosto 25, 2010.
- [75] *“BIOS”*
Autor: Wikipedia.
<http://es.wikipedia.org/wiki/BIOS>
Última visita: noviembre 22, 2010.
- [76] *“Eucalyptus Network Configuration (1.6)”*
Autor: Eucalyptus Systems, Inc.
http://open.eucalyptus.com/wiki/EucalyptusNetworking_v1.6
Última visita: julio 30, 2010.
- [77] *“Tool for the cloud: Elasticfox”*
Autor: VANOVER, Rick.
<http://blogs.techrepublic.com.com/networking/?p=1533>
Última visita: septiembre 12, 2010.
- [78] *“Complemento, plug-in”*
Autor: Wikipedia.
http://es.wikipedia.org/wiki/Complemento_%28inform%C3%A1tica%29
Última visita: septiembre 5, 2010.
- [79] *“Amazon Elastic Compute Cloud-Getting Started Guide”*
Autor: ONO, Hiroshi.
<http://www.slideshare.net/rawwell/elasticfox-owners-manual-presentation>
Última visita: noviembre 11, 2010.
- [80] *“¿Qué son el Cloud Computing y Google App Engine?”*
Autor: DÍAZ, Raúl.
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=cloudcomputing>
Última visita: noviembre 11, 2010.
- [81] *“Dual Core”*
Autor: Wikipedia.
<http://es.wikipedia.org/wiki/Dual-core>

Última visita: noviembre 22, 2010.

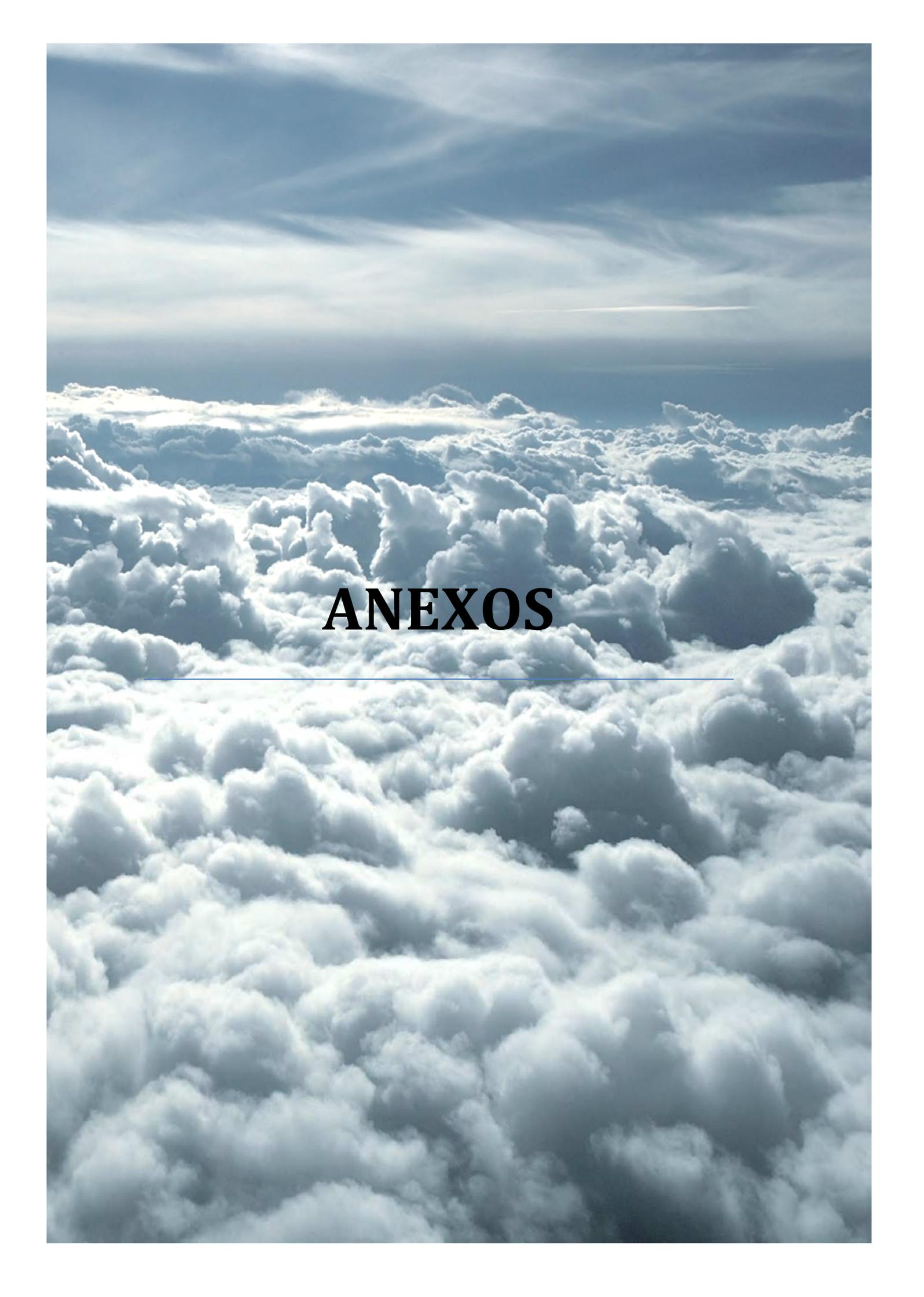
- [82] *"Understanding Eucalyptus database schema"*
Autor: Wikipedia.
<http://community.eucalyptus.com/forum/understanding-eucalyptus-database-schema>
Última visita: noviembre 29, 2010.
- [83] *"Raw Device"*
Autor: Wikipedia.
http://en.wikipedia.org/wiki/Raw_device
Última visita: noviembre 29, 2010.
- [84] *"Installation From Linux"*
Autor: LEE, Curtis.
<https://help.ubuntu.com/community/Installation/FromLinux>
Última visita: noviembre 29, 2010.
- [85] *"Installing LAMP on Ubuntu for Newbies"*
Autor: Cargoship.
http://www.howtoforge.com/ubuntu_lamp_for_newbies
Última visita: noviembre 29, 2010.
- [86] *"Euca2ools User Guide"*
Autor: Eucalyptus Systems Inc.
http://www.howtoforge.com/ubuntu_lamp_for_newbies
Última visita: diciembre 7, 2010.
- [87] *"Amazon EC2 AMI Tools"*
Autor: Amazon Web Services LLC.
http://aws.amazon.com/developertools/368?_encoding=UTF8&jiveRedirect=1
Última visita: diciembre 7, 2010.
- [88] *"Amazon EC2 API Tools"*
Autor: Amazon Web Services LLC.
http://aws.amazon.com/developertools/351?_encoding=UTF8&jiveRedirect=1
Última visita: diciembre 7, 2010.
- [89] *"UECCDInstall"*
Autor: KIRKLAND, Dustin
<https://help.ubuntu.com/community/UEC/CDInstall>
Última visita: diciembre 18, 2010.
- [90] *"IPTraf, estadísticas sobre el tráfico de tu red"*
Autor: Dragon, JAR
<http://www.dragonjar.org/iptraf-estadisticas-sobre-el-trafico-de-tu-red.xhtml>
Última visita: diciembre 26, 2010.
- [91] *"iftop: display bandwidth usage on an interface"*
Autor: WARREN Paul, LIGHTFOOT Chris
<http://www.ex-parrot.com/pdw/iftop/>
Última visita: diciembre 26, 2010.

[92] *“Amazon Web Services Introduces Cluster Compute Instances, an Amazon EC2 Instance Type Tailored for High Performance Computing”*

Autor: Desconocido

<http://hotcellularphone.com/breaking-news/amazon-web-services-introduces-cluster-compute-instances-amazon-ec2-instance-type-tailored-high-performance-computing/>

Última visita: enero 06, 2010.



ANEXOS

ANEXO A: PROVEEDORES DE CLOUD COMPUTING

Feature	Amazon Web Service	GoGrid	Flexiscale	Mosso
Computing Architecture	<ul style="list-style-type: none"> -Elastic Compute Cloud (EC2) allows uploading XEN virtual machine images to the infrastructure and gives client APIs to instantiate and manages them. -Public Cloud 	<ul style="list-style-type: none"> -Data center architecture which is designed to deliver a guaranteed QoS level for the exported services. -Autonomically reconfiguring for infrastructure to cater for fluctuations in the demand 	<ul style="list-style-type: none"> -Data center architecture which is designed to deliver a guaranteed QoS level for the exported services. -Autonomically reconfiguring for infrastructure to cater for fluctuations in the demand -Allows multi-tier architectures through a high-speed internal GigE network -Public Cloud 	<ul style="list-style-type: none"> -Merge the idea of cloud computing with the traditional managed/shared server environment that many web hosts provide -Doesn't provide root Access to their servers -Servers are in intelligent clusters, which make us fairly efficient with infrastructure and power usage -Public Cloud
Virtualization Management	-OS level running on a Xen hypervisor	-Xen hypervisor	-Xen hypervisor	-VMware ESX Server
Service	IaaS, Xen images	IaaS	IaaS	IaaS
Load balancing	<ul style="list-style-type: none"> -Service will allow users to balance incoming requests and traffic across multiple EC2 instances -Round-robin load balancing, HAProxy 	<ul style="list-style-type: none"> -F5 load balancing, Load balancing algorithm used as Round Robin, Sticky Session, SSL Least Connect, Source address 	<ul style="list-style-type: none"> -Automatic Equalization of Server Load within Cluster 	<ul style="list-style-type: none"> -Scales with traffic, inherits load-balancing

Fault tolerance	-System should automatically alert, failover, and re-sync back to the "last known state" as if nothing had failed	-Instantly scalable and reliable file-level backup service	-Full self-service: Start/stop/delete, change memory/CPU/storage/IPs of Virtual Dedicated Servers	-Auto managing
Interoperability	-Support horizontal Interoperability e.g. interoperability among EC2, Eucalyptus etc	-Committed to furthering interoperability	-Releasing some code that will work for interoperability with other platforms to promote standardization	-Open Cloud Manifest
Storage	-Simple Storage Service (S3) and Simple DB. Simple DB provides a semi-structured data store with querying capability.	-Storage is a two-step process i) Connecting each server to Private Network ii) transfer protocols (RSYNC, FTP, SAMBA, SCP to transfer data to and from Cloud Storage)	-Persistent storage based on a fully virtualized high-end SAN/NAS back-end	-Storage is based on Rackspace Mosso's Cloud Files which is reliable, scalable and affordable web-based storage for backing up and archiving all user static content.
Security	-Type II (SAS70 Type II) certification, firewall, X.509 certificate, SSL-protected API, Access Control List	-Don't provide a guarantee of security	-Customer has own VLAN, Virtual Dedicated Servers with SLA and Tier 1 top quality storage Backend	-Payment Card Industry Data Security Standard
Programming Framework	-Amazon Machine Image (AMI), Amazon Map reduce framework	-Uses a REST-like Query interface & GoGrid API supports Java, Python, Ruby	-Flexiscale API support C, C#, C++, Java, PHP, Perl and Ruby	-Currently supports ASP, PHP

Fuente: [53]

**Tabla A.1 Cloud Computing Infrastructure
Technology and Solution Provider**

Feature	Google App Engine	GigaSpaces	Azure	RightScale	SunCloud	Force.com
Computing Architecture	-Google's geo-distributed architecture	-Space base architecture & used for mission-critical applications, where the need for extreme performance, reliability an alternative to traditional tier-based architectures	-An internet-scale cloud services platform hosted in Microsoft data centers, which provides an OS and a set of developer services that can be used individually or together	-Multi-server clusters -Gives Virtual Private Servers monitoring system is one of the advanced features of the RightScale Dashboard -Dynamic server configuration -Elastic IPs -Single worker processing stage	-Solaris os, and Zettabyte File System (ZFS) -Clusters of servers and public IP addresses -Hybrid Cloud -Q-layer enabled for Data Warehouse and Enterprise Resource Planning -Open Dynamic Infrastructure Management Strategy	-Multitenant Architecture with metadata-driven development model -Allowing a single application to serve multiple customers
Virtualization Management	-Multitenant Architecture	-Application-level Virtualization	-Hypervisor (base Hyper-V)	-Xen hypervisor	-Hypervisor (Sun xVM Server), OS (Solaris Conainers), network (Crossbow), storage (COMSTAR, ZFS), and application (GlassFish & Java CAPS Technologies).	-Everyone stars with the same version of the application. The data are stored in a shared database, but each client has Access only

						to its own information
Service	PaaS Runtime interpreter	PaaS	PaaS	PaaS	PaaS	SaaS Confined to API
Load balancing	-Automatic scaling and load balancing	-Performed through the GigaSpaces high-performance communication protocol the EC2 network infrastructure	-Built-in hardware load balancing	-High Availability Proxy load balancing in the cloud	-Hardware balancers may outperform software based balancers	-Load balancing among tenants
Fault tolerance	Automatically pushed to a number of fault tolerant servers -App Engine Cron Service	-Each node in the AMI cluster automatically discovers the other nodes and the cluster becomes a fault-tolerant cluster	-Containers are used for load balancing and availability: If a failure occurs, SQL Data Services will automatically begin using another replica of the container	-Basic, intermediate & advance Failover Architectures as described in Best Practices for using Elastic IPs (EIP) and Availability Zones	-Resource based scheduling of service request & automatic failover when node fails	-Self management and self-tuning
Storage	-Proprietary database (Big Table distributed-storage)	-In memory data grid object relation mapping support -Support Amazon Security Groups	-SQL Server Data Services (SSDS) -Allows storing binary large objects (blobs) - Security	- Open storage model, MySQL backups are Elastic Block Store (EBS) are saved to S3	-MySQL's row-based replication -Routine use of backup with replication	-Force.com database deals in terms of relationship fields

<p>Security</p>	<p>-Google Secure Data Connector -SDC uses RSA/128-bit or higher AES CBC/SHA -SDC uses TLS-based server authentication</p>	<p>-Built-in SSH tunneling</p>	<p>-Token service (STS) creates Security Assertion Markup Language token according to rule</p>	<p>-Assign Multiple Security Groups to an Instance</p>	<p>-User-provisioning and meta directory solution -Process and user rights management trusted extensions for mandatory Access Control -Control and role-based Access management to back-line resources</p>	<p>-SysTrust SAS 70 Type II -Users and security, programmatic and platform security framework</p>
<p>Programming Framework</p>	<p>-MapReduce programming framework that support Python, Java as well as several java related standards such as the Java Servlet API, JDO and JPA</p>	<p>-Non intrusive programming model, single cluster model supports for String/Java .NET, C++</p>	<p>-Microsoft .NET</p>	<p>-Ruby Amazon's Simple Queue Service</p>	<p>-Solaris OS, Java, C, C++, FORTRAN, Open API-Public, RESTful, Java, Python, Ruby</p>	<p>-Apex language for database service and supports for .NET, C# Apache Axis (Java and C++)</p>

Fuente: [53]

Tabla A.2 Cloud Computing PASS and SAAS provider

Feature	Eucalyptus	OpenNebula	Nimbus	Enomaly
Computing Architecture	<ul style="list-style-type: none"> -Ability to configure multiple clusters, each with private internal network addresses, into a single Cloud. -Private Cloud 	<ul style="list-style-type: none"> -Cluster into an IaaS cloud -Focused on the efficient, dynamic and scalable management of VMs within datacenters (private clouds) involving a large amount of virtual and physical servers -Based on Haizea scheduling 	<ul style="list-style-type: none"> -Science cloud -Client-side cloud-computing interface to Globus-enabled TeraPort cluster -Nimbus Context Broker that combines several deployed virtual machines into "turnkey" virtual clusters -heterogeneous clusters of auto-configuring VMs with one command 	<ul style="list-style-type: none"> -A clustered virtual server hosting platform; ElasticDrive, a distributed remote storage system; and GeoStratus, a private content delivery network. -Uses a clustered file-system called GlusterFS and is capable of scaling to several petabytes. It aggregates various storage bricks over Infiniband, RDMA or TCP/IP interconnects into one large parallel network file system -Private Cloud
Virtualization Management	<ul style="list-style-type: none"> -Xen hypervisor 	<ul style="list-style-type: none"> -Xen, KVM and on-demand Access to Amazon EC2 	<ul style="list-style-type: none"> -Xen Virtualization 	<ul style="list-style-type: none"> -Kernel-based virtual machine (KVM) supports Xen Open VZ and Sun's VirtualBox
Service	IaaS	IaaS	IaaS	IaaS
Load balancing	<ul style="list-style-type: none"> -Simple load-balancing cloud controller 	<ul style="list-style-type: none"> -Nginx Server configured as load balancer, used round-robin or weighted selection mechanism 	<ul style="list-style-type: none"> -Launches self-configuring virtual clusters i.e. the context broker 	<ul style="list-style-type: none"> -Uses user-mode load-balancing software with its own network stacks that runs over Linux and Solaris in the form of a virtual server -Supports different load balancing methods, including round robin, random, hash, and least resource

Fault tolerance	-Separate cluster within the Eucalyptus cloud reduce the chance of correlated failure	-The daemon can be restarted and all the running VMs recovered -Persistent database backend to store host and VM information	-Checking worker nodes periodically and recovery	-Overflow, disaster and failover services
Ineroperability	-Multiple cloud computing interfaces using the same "back end" infrastructure	-Interoperable between intra cloud services	-Standards: "rough consensus and working code"	-Cloud portability and interoperability to cross cloud vendors
Storage	-Walrus (the front end for the storage subsystem)	-Database, persistent storage for ONE data structures -SQLite3 backend is the core component of the OpenNebula internal data structures	-GridFTP and SCP	-Multiple remote cloud storage services (S3, Nivanix and CloudFS)
Security	-WS-security for authentication, Cloud controller generates the public/private key	-Firewall, Virtual Private Network Tunnel	-PKI credential required -Works with Grid proxies VOMS, Shibboleth (via GridShib), custom PDPs	-Clustered handling of security
Programming Framework	-Hibernate, Axis2 and Axis2c, Java	-Java, Ruby	Python, Java	-Ruby on rails, PHP, Python

Fuente: [53]

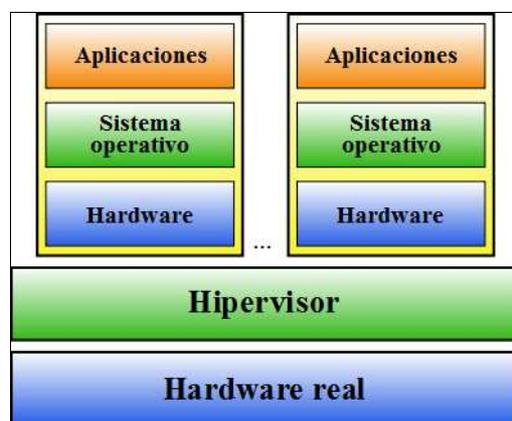
Tabla A.3 Open Source Based Cloud Computing Services

ANEXO B: VIRTUALIZACIÓN

B.1 VIRTUALIZACIÓN DE LA PLATAFORMA

Involucra la simulación de una máquina real, la cual está alojada en un sistema anfitrión (*host*) y se ejecuta a través de un software llamado *hypervisor*⁸³, el cual controla que todas las máquinas virtuales sean atendidas correctamente. El *hypervisor* crea una capa de abstracción entre el hardware de la máquina física (*host*) y el sistema operativo de la máquina virtual (*virtual machine guest*) de tal forma que maneja los recursos de las máquinas físicas subyacentes para que el usuario pueda crear varias máquinas virtuales presentando a cada una de ellas, una interfaz del hardware que sea compatible con el sistema operativo elegido. [14]

Existen dos tipos de *hypervisors*: *Non Hosted* (Figura B.1), que corre directamente sobre el hardware físico de la máquina real, y el denominado *Hosted* (Figura B.2), el cual corre sobre un sistema operativo anfitrión [47].



Fuente: [48]

Figura B.1 *Hypervisor del tipo Non Hosted*

⁸³ **Hypervisor o VMM (Virtual Machine Monitor)** es una plataforma de virtualización que permite utilizar, al mismo tiempo, diferentes sistemas operativos o máquinas virtuales (sin modificar o modificados) en una misma computadora central.



Fuente: [4]

Figura B.2 Hypervisor del tipo Hosted

Tipos de virtualización de plataforma:

Dentro de este tipo de virtualización se encuentran tres subdivisiones de las cuales es particularmente importante, la *virtualización de equipos y servidores*, para la realización de este proyecto.

B.1.1 VIRTUALIZACIÓN DE APLICACIONES

“Las aplicaciones virtuales se ejecutan en un pequeño entorno virtual que actúa como una capa entre la aplicación y el sistema operativo, eliminando los conflictos entre aplicaciones y entre aplicaciones y el sistema operativo”. [15]

B.1.2 VIRTUALIZACIÓN DE ESCRITORIO

El escritorio virtualizado es almacenado remotamente en un servidor central en lugar de almacenarlo en el disco duro local. Los usuarios trabajan en su escritorio remoto desde su PC (*Personal Computer*), portátil o un dispositivo móvil, *smartphone* o cliente ligero (*thin client*)⁸⁴.

⁸⁴ **Thin client** depende principalmente del servidor central para las tareas de procesamiento, y principalmente se enfoca en transportar la entrada y la salida entre el usuario y el servidor remoto.

B.2 VIRTUALIZACIÓN DE EQUIPOS Y SERVIDORES

Permite ejecutar varias máquinas virtuales en un solo equipo anfitrión (*host*), ofrece disponibilidad de diferentes servidores dentro de una misma máquina, abstrayendo a los servidores virtuales del hardware real permitiendo asignarles los recursos necesarios, o incluso pudiendo moverlos de un servidor físico a otro, en tiempo real.

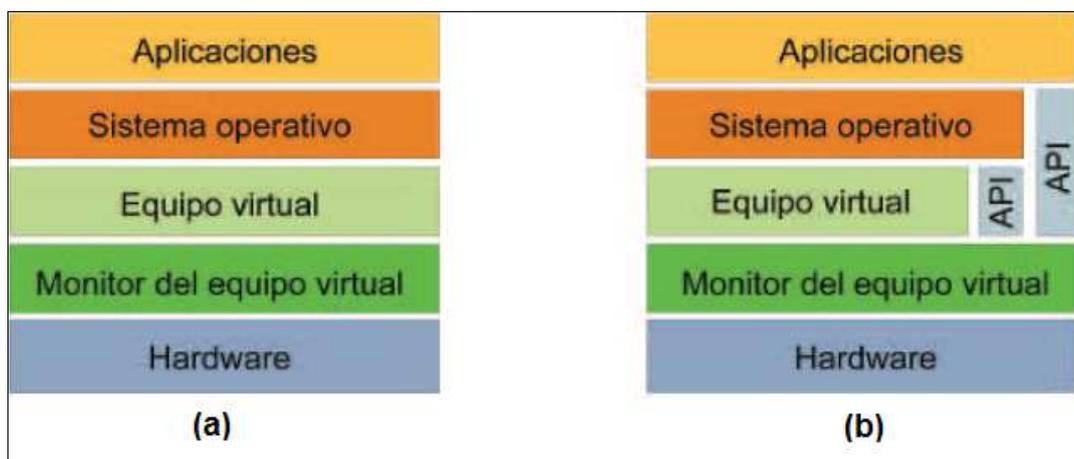
La virtualización de servidores se divide en:

B.2.1 VIRTUALIZACIÓN COMPLETA

Es una técnica que permite ejecutar sistemas operativos huésped simulando todo el hardware del equipo anfitrión (*host*), sin tener que modificarlos, y utilizando en medio un *hypervisor* que permite compartir el hardware real. [16]

Su principal ventaja es que los sistemas operativos pueden ejecutarse sin ninguna modificación sobre la plataforma, pero éstos deben ser soportados por una arquitectura virtualizada. [16]

La virtualización completa permite que el hardware simulado de cada máquina virtual sea independiente del equipo anfitrión.



Fuente: [47]

Figura B.3 Virtualización Completa vs Paravirtualización

Como se puede apreciar en la Figura B.3 a, los sistemas operativos huéspedes (*guest*) corren en la máquinas virtuales ejecutadas sobre el monitor del equipo virtual (*hypervisor* o *VMM*) el cual se comunica directamente con el hardware, el tipo de *hypervisor* que se especifica es del tipo *Non Hosted*.

Ejemplos de plataformas que ofrecen virtualización completa: KVM, VMware Workstation.

B.2.2 PARAVIRTUALIZACIÓN

La paravirtualización surgió como una forma de mejorar la eficiencia de las máquinas virtuales y acercarlo al rendimiento del hardware nativo. Por eso, los sistemas virtualizados huésped, deben estar basados en sistemas operativos especialmente modificados para ejecutarse sobre un *hypervisor*. De esta forma no es necesario que éste monitorice todas las instrucciones, sino que los sistemas operativos huésped y anfitrión colaboren en la tarea. [16]

El modelo de paravirtualización (Figura B.3 b) incluye APIs que asisten al monitor del equipo virtual. [47]

En la virtualización completa, todo el sistema se emula (BIOS, disco, etc.), pero en la paravirtualización, su módulo de gestión opera con un sistema operativo que ha sido adaptado para trabajar en una máquina virtual. La paravirtualización suele funcionar mejor que el modelo de virtualización completa, simplemente porque en una implementación totalmente virtualizada (virtualización completa), todos los elementos deben ser emulados, a diferencia de la paravirtualización en la que solo el hardware subyacente es emulado. [47], [17]

Ejemplos de paravirtualización: Xen, VMware ESX

En la Tabla B.1 se presenta una comparación entre Virtualización Completa y Paravirtualización.

Tipo de Virtualización	Instancias	Gastos	Necesidades de Sistema de Procesamiento	Total
Virtualización Completa	5	10% (50% total)	10% (50% total)	100%
Paravirtualización	8	2% (16% total)	10% (50% total)	96%

Fuente: [47]

Tabla B.1 Potencia del procesador utilizado en la virtualización completa y paravirtualización

B.2.3 VIRTUALIZACIÓN HÍBRIDA

O también conocida como Virtualización Nativa. *“Éste es el más reciente tipo de virtualización orientado a la virtualización de la arquitectura x86. Constituye una combinación de la virtualización completa o de la paravirtualización en conjunto con técnicas de aceleración de I/O y memoria. Permite sacar ventaja de las últimas tecnologías de la arquitectura x86, Intel VT y AMD-V (Mayor información Anexo B.3) diseñadas para la virtualización asistida por hardware”.* [47]

Su funcionamiento es similar al de la virtualización completa, es así que los sistemas operativos pueden ser instalados sin una modificación previa. [16]

“La virtualización nativa emplea técnicas de aceleración de I/O y memoria, de manera selectiva de acuerdo a la necesidad. Soporta arquitectura de 64 para ciertos sistemas operativos; tiene el mayor de CPU, memoria, y I/O de todos los tipos de virtualización x86. Sin embargo requiere de una arquitectura de CPU capaz de soportar aceleración asistida de hardware (Intel I/OAT)”. [47]

B.3 EXTENSIÓN DE VIRTUALIZACIÓN

B.3.1 TECNOLOGÍA DE VIRTUALIZACIÓN INTEL (INTEL-VT)

Es un conjunto de mejoras de hardware para plataformas de servidores y clientes Intel que proporcionan soluciones de virtualización basadas en software. [69]

Permite que una plataforma de hardware funcione como múltiples plataformas virtuales, es decir proporciona la utilización máxima de un sistema mediante la consolidación de entornos múltiples en un único servidor o PC. [70], [71]

Ofrece mayor capacidad de gestión, limita el tiempo de inactividad y conserva la productividad de los empleados al aislar las actividades informáticas en particiones independientes. [70]

B.3.2 VIRTUALIZACIÓN AMD (AMD-V)

Este tipo de virtualización incluye un conjunto de extensiones de hardware a la arquitectura de sistema x86 que permite obtener mayores ventajas de los recursos que se manejan, lo que aumenta la eficiencia de los servidores, clientes y centros de datos. [72]

Mejora de una serie de características de diseñadas para aumentar el rendimiento, la fiabilidad y la seguridad de los entornos de virtualización actuales y futuros al permitir aplicaciones virtualizadas con acceso directo y rápido a su memoria asignada. [73]

Se tiene como una ventaja que contribuye a una ejecución más segura y eficaz del software de virtualización al permitir una mejor experiencia con los sistemas virtuales, ya que simplifica las soluciones de virtualización, para ofrecer una experiencia más satisfactoria al usuario y un mejor rendimiento de aplicaciones. [72], [73]

La tecnología AMD-V proporciona las siguientes características:

Función que permite acelerar el rendimiento de muchas de las aplicaciones virtualizadas mediante la gestión de memoria de los equipos virtuales basada en hardware. [73]

“Extensiones de la virtualización al conjunto de instrucciones x86: un software que crea equipos virtuales con mayor eficacia, de modo que los distintos sistemas operativos y sus aplicaciones se puedan ejecutar simultáneamente en el mismo ordenador”. [73]

TLB etiquetado: características de hardware que facilitan el cambio eficaz entre los equipos virtuales para una mejor respuesta de las aplicaciones. [73]

AMD-V™ Extended Migration: característica de hardware que ayuda a que el software de virtualización lleve a cabo una migración en directo de los equipos virtuales entre todas las generaciones de procesadores AMD disponibles. [73]

Virtualización de E/S: solución que permite el acceso directo al dispositivo a través de un equipo virtual, omitiendo el hypervisor para mejorar el rendimiento de la aplicación y el aislamiento de los equipos virtuales, con objeto de incrementar la integridad y la seguridad. [73]

B.4 HYPERVISORS

B.4.1 XEN

Xen es una máquina virtual de código abierto desarrollada por la Universidad de Cambridge. Se ejecuta directamente sobre el hardware y trata incluso al sistema operativo anfitrión como una máquina virtual de los clientes.

Es un *hypervisor* de código abierto que se ejecuta directamente en el hardware, permitiendo que múltiples sistemas operativos se ejecuten al mismo tiempo y en el mismo hardware, esto lo hace sin emulación o traducción de instrucciones, y de forma proporciona al de rendimiento del CPU.

Los sistemas operativos pueden ser modificados explícitamente para correr Xen (aunque manteniendo la compatibilidad con aplicaciones de usuario). Esto permite a Xen alcanzar virtualización de alto rendimiento sin un soporte especial de hardware.

Intel ha realizado diversas contribuciones a Xen que han permitido añadir soporte para sus extensiones de arquitectura VT-X *Vanderpool*. Esta tecnología permite que sistemas operativos sin modificar actúen como hosts dentro de las máquinas virtuales de Xen, siempre y cuando el servidor físico soporte las extensiones VT de Intel o Pacifica de AMD. [54]

Xen está optimizado para los servidores, ejecutando varios sistemas operativos Linux u otros, cada uno con su propio núcleo, de manera segura y limpia.

B.4.2 KVM

KVM (Kernel-based Virtual Machine)

KVM es el software de código abierto, que representa una solución de virtualización completa para arquitecturas de hardware x86 que contengan extensión de virtualización (Intel-VT ó AMD V).

Con KVM se puede ejecutar múltiples máquinas virtuales que ejecutan imágenes de Linux o de Windows sin modificar. Cada máquina virtual tiene hardware virtualizado, una tarjeta de red, disco, adaptador de gráficos, etc. El componente del *kernel* de KVM está incluido en la línea principal de Linux, a partir del 2.6.20.

KVM es un módulo más del *kernel*. Esto significa que no es un *kernel* diferente como le ocurre a Xen, al ser un módulo del *kernel*. [55]

Se lo puede cargar y descargar en tiempo de ejecución, viene por defecto del *kernel*.

Debido a que es parte del *kernel*, “hereda”, es decir usa muchas de las ventajas que ya tiene el *kernel*: [55]

- Escalabilidad
- Soporte para diferentes procesadores (Itanium, x86_64 y x86)

- Gestión de memoria

Otra ventaja que se puede mencionar es que, al ser parte del *kernel*, tiene menos líneas de código ya que muchas características las aprovecha del propio *kernel*. Esto implica un desarrollo más fácil y centralizado. [55]

B.4.3 QEMU

Qemu es un emulador de máquinas virtuales que maneja una traducción binaria dinámica, ésta permite la emulación de un conjunto de instrucciones mediante la traducción de código. Mediante el uso de traducción dinámica, se logra un alto rendimiento. [56], [57], [58]

Maneja el paquete *qemu-launcher*, una interfaz gráfica para hacer su utilización más amigable (ya que está pensado para ser ejecutado y administrado desde la consola), hace uso de GTK+⁸⁵ por lo que se integra muy bien con Gnome. Además del módulo acelerador *kqemu* para conseguir una velocidad próxima a la nativa en sistemas i386. [60]

Cuando se utiliza Qemu como un emulador de máquinas, éste puede correr sistemas operativos y programas hechos para una sola máquina en equipos diferentes. [58]

Soporta virtualización al ejecutarse bajo el *hypervisor* Xen o usando el módulo del *kernel* KVM en Linux. [58]

En relación con la emulación de la CPU, proporciona un conjunto de modelos de equipos, permitiendo la ejecución de una variedad de sistemas operativos sin modificar, por lo que esto puede ser visto como un monitor de máquina virtual alojada. [56]

⁸⁵ **GTK+** es una biblioteca para crear interfaces gráficas de usuario (GUI). [59]

También puede ser utilizado exclusivamente para la emulación de la CPU para los procesos a nivel de usuario. [56]

Una característica exclusiva de Qemu es la portabilidad: las máquinas virtuales pueden ejecutarse en cualquier computador, incluso en aquellos en los que el usuario tiene derechos limitados, es decir que no tienen acceso de administrador. [56]

De acuerdo a la licencia que maneja Qemu, se puede decir que:

Qemu en su conjunto se distribuye bajo GNU *General Public License*. [58]

Partes de Qemu tienen licencias específicas que sean compatibles con la GNU *General Public License*. Por lo tanto cada archivo de origen contiene su propia información de licencias. Muchas fuentes de emulación de dispositivos de hardware son liberados bajo la licencia BSD (*Berkeley Software Distribution*). [58]

En relación con la implementación de *Cloud Computing* utilizando *Ubuntu Enterprise Cloud* y *Eucalyptus*, se puede decir que mientras el rendimiento sea bueno, *Eucalyptus* puede fácilmente usar Qemu en vez de KVM para la virtualización sin extensiones de hardware virtualizado. [61]

B.4.4 VMWARE

VMware es un sistema de virtualización por software, es decir es un programa que simula un sistema físico (un ordenador, un hardware) con unas características de hardware determinadas. Cuando se ejecuta el programa (simulador), proporciona un ambiente de ejecución similar a todos los efectos a un ordenador físico (excepto en el puro acceso físico al hardware simulado), con CPU (puede ser más de una), BIOS, tarjeta gráfica, memoria RAM, tarjeta de red, sistema de sonido, conexión USB, disco duro (pueden ser más de uno), etc. [62]

Un virtualizador por software permite simular varios sistemas operativos dentro de un mismo hardware de manera simultánea, permitiendo así el mayor aprovechamiento de recursos. No obstante, y al ser una capa intermedia entre el sistema físico y el sistema operativo que funciona en el hardware emulado, la velocidad de ejecución de este último es menor, pero en la mayoría de los casos es suficiente para usarse en entornos de producción. [62]

VMware ofrece las mejores soluciones y mayor confianza para transformar entornos de TI en infraestructuras flexibles, infraestructuras automatizadas de Nube, ayudando a que el cliente obtenga ahorros de costes y beneficios de productividad. [63]

ANEXO C: DUAL CORE

El presente anexo ha sido citado de la referencia. [81]

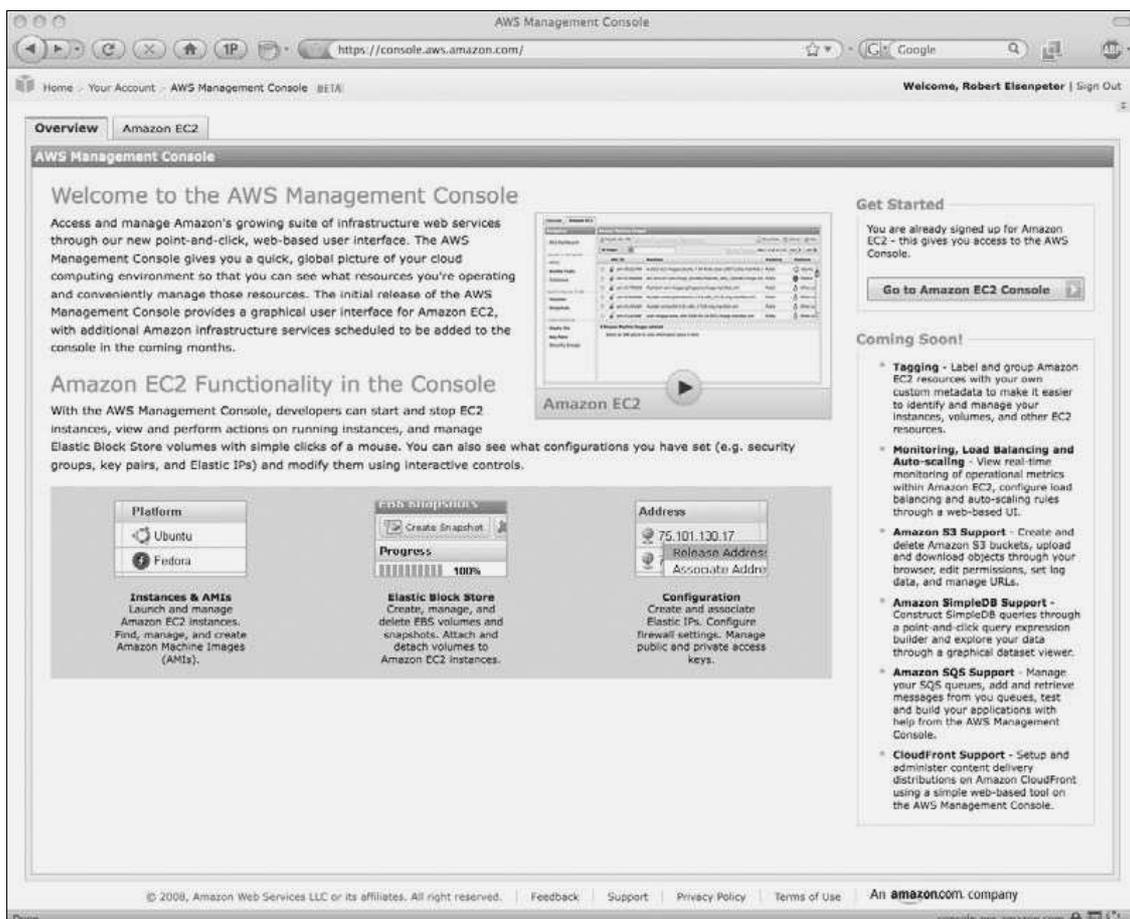
Equivalente de doble núcleo. Término relacionado con el hallazgo de dos núcleos o CPU en un mismo chip de procesador, tanto en el mismo encapsulado de silicio como en silicios separados aunque dentro del mismo chip.

Existen dos núcleos idénticos en un mismo circuito integrado o chip, trabajando a la misma velocidad, aunque pudiendo ajustarse cada uno según la carga y controlador que lo gobierne. Por defecto, si no se le indica bajo un *kernel* de UNIX/Linux o no se le instalan controladores bajo Windows, trabajan al máximo rendimiento. En el caso de Linux, el demonio ACPID puede ajustar automáticamente la tasa de la CPU para bajar el consumo/calor generado, pero esto puede deshabilitarse tanto por un nuevo *kernel* como por el uso de `cpufreq-select`. En el caso de otros sistemas UNIX, como BSD, la tasa lo ajusta automáticamente el demonio `powerd`.

La aparición del doble núcleo redujo la velocidad punta en cada uno de ellos, (por ejemplo, un núcleo sencillo de 3GHz fue reemplazado por un núcleo dual de 2,2GHz x2), pero esta reducción podría no verse afectada directamente en el rendimiento, ya que depende del tipo de núcleo de CPU que tenga instalado, así como el nivel de caché y velocidad de FSB. También importa, como se comenta en el siguiente punto, si la aplicación soporta el trabajo conjunto (en paralelo) con varias CPU y si el sistema operativo reparte bien la faena. Más adelante, la frecuencia de reloj fue aumentando, hasta sobrepasar los 3GHz por núcleo.

ANEXO D: AMAZON WEB SERVICES (AWS)

Grandes compañías están peleando por escalar posiciones dentro del desarrollo de *Cloud Computing*. Por ejemplo *Amazon*, ofrece varios servicios de Nube a través de Internet, que van desde el almacenamiento hasta plataformas de bases de datos, estos servicios se pueden administrar a través de una consola de administración, como se muestra en la Figura D.1.



Fuente [16]

Figura D.1 AWS Management Console

Amazon Web Services (AWS) que es una colección de Servicios Web, que conforman una plataforma de *Cloud Computing* ofrecida por *amazon.com*. Sus servicios centrales y más conocidos son *Amazon Elastic Compute Cloud (EC2)* y *Amazon Simple Storage Service (S3)*. [16], [18]

Amazon Web Services (AWS) ofrece servicios online para otros sitios web, muchos de estos servicios no son mostrados directamente a los usuarios finales, pero se ofrece la funcionalidad de que otros desarrolladores puedan usarlos. [65]

Entre los usos más comunes que los clientes le dan a *Amazon Web Services* (AWS) podemos enumerar:

Para los clientes con una necesidad continua de mover rápidamente grandes volúmenes de datos dentro o fuera de la nube de AWS, *Amazon* permite la interconexión entre redes. [16]

Hosting de aplicaciones, los proveedores de aplicaciones ya sea bajo el modelo *Software as a Service* (SaaS) o *Application Service Provider* (ASP) están utilizando la infraestructura de *Amazon* para albergar las aplicaciones que dan servicio a sus clientes, es decir se ofrece *Infrastructure as a Service* (IaaS). Tiene mucha lógica que los otros servicios que ofrece *Cloud Computing*, *Platform as a Service* (PaaS) y *Software as a Service* (SaaS), utilicen la infraestructura de *Amazon* para dar cobertura a todas sus demandas. [65]

Almacenamiento y copia de seguridad, aunque el coste del hardware para almacenamiento es cada vez más bajo, los costes para mantenimiento y aprovisionamiento siguen estando ahí. Si se suma esto al coste bajo, fiabilidad y escalabilidad del servicio *Amazon Simple Storage Service* (*Amazon S3*), que utiliza una interfaz de Servicios Web simple, permite almacenar cualquier cantidad de datos desde cualquier lugar de la Web. [16], [65]

Supercomputación, el servicio *Amazon Elastic Compute Cloud* (*Amazon EC2*) provee el acceso a una gran cantidad de computación y muchas empresas lo están utilizando para realizar un gran número de operaciones de cómputo que procesan grandes cantidades de información. [65]

Entrega de contenido, por ejemplo si se desea entregar contenido a clientes o empleados en diferentes posiciones geográficas del mundo y de forma inmediata, es decir, sin latencias provocadas por la localización del contenido y con altas tasas de

transferencia, se puede utilizar *Amazon Cloud Front*, que es un servicio de este tipo con un rendimiento óptimo. [66]

Almacenamiento y distribución de contenido multimedia, esto puede ser un negocio impredecible y costoso. *Amazon Web Services (AWS)* permite su distribución rentable, escalable y confiable con *Amazon Simple Storage Service (Amazon S3)* y *Amazon Cloud Front*. [19]

Base de Datos, *Amazon* dispone de dos servicios de base de datos: *Amazon SimpleDB* y *Amazon RDS*. *SimpleDB* permite crear una base de datos sencilla y con *RDS* puedes crear bases de datos relacionales. [66]

Hay que considerar que *Amazon* ha desarrollado una Nube pública, mientras que *Eucalyptus* permite el desarrollo de Nubes privadas, éstas últimas tienen la posibilidad de comunicarse con la Nube pública de *Amazon*, por lo cual se pueden ofrecer servicios compatibles entre sí, como:

Block Storage Service en *Eucalyptus* es la interfaz compatible con *Elastic Block Store EBS* de *Amazon*, por lo tanto, se puede utilizar comandos de *Amazon EC2* para interactuar con *Block Storage Service* de *Eucalyptus*, solo se deben descargar e instalar las herramientas *EC2 API* para poder usar el bloque de almacenamiento. [67]

Walrus es un servicio de almacenamiento que maneja *Eucalyptus*, éste es compatible con la interfaz de *Simple Storage Service S3* de *Amazon*. [68]

Elastic Cloud Compute EC2 de *Amazon* es un servicio web que proporciona capacidad de cómputo variable en la Nube.

D.1 EC2TOOLS

Ec2ools son las herramientas de línea de comandos oficiales de *Amazon* y son utilizadas por los usuarios para manejar los Servicios Web de *Amazon EC2*.

Al igual que *euca2ools*, *ec2ools* deben ser instaladas por el cliente para poder hacer uso de las mismas, éste necesita tener las *ec2-ami-tools* y *ec2-api-tools*.

ec2-ami-tools son herramientas que permiten la inserción de imágenes AMI (*Amazon Machine Image*), crear una AMI de una maquina virtual o volumen existente, y cargar una AMI a *Amazon S3*. [87]

ec2-api-tools son herramientas que permiten el registro y puesta en marcha de instancias, la manipulación de grupos de seguridad, etc. [88]

ANEXO E: CONFIGURACIÓN DE LA BIOS EN LOS NODOS

“El BIOS (Basic Input/Output System) es un software, instalado en la placa base, que se ejecuta al encender la computadora. Se encarga de localizar y reconocer todos los dispositivos necesarios para cargar el sistema operativo en la memoria RAM.

De acuerdo a las características del procesador (Intel Q8400, Figura E.2) instalado en la placa madre de los nodos (Intel DG-35EC), se tiene el soporte para Extensión de Virtualización el cual requiere de activación a través de la BIOS, si no se configura esta característica el sistema operativo no será capaz de detectarla. En la Figura E.1 se muestra la pantalla para la configuración respectiva.” [75]

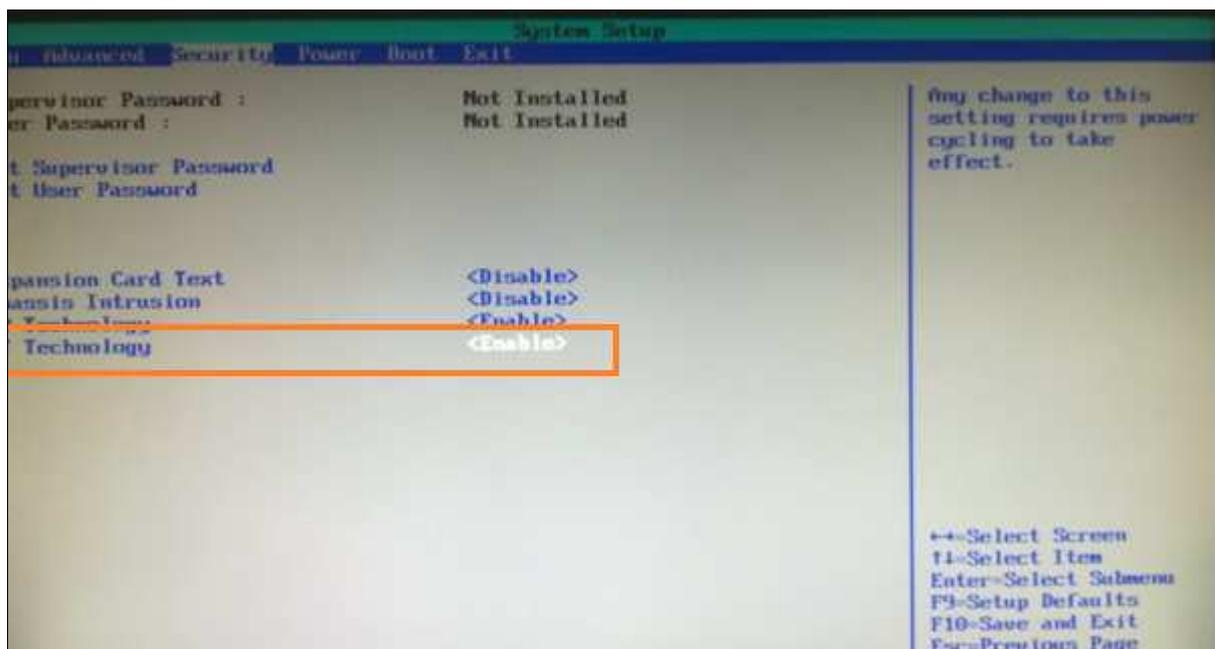


Figura E.1 Activación de extensión de virtualización en la BIOS

Intel® Core™2 Quad Processor Q8400 (4M Cache, 2.66 GHz, 1333 MHz FSB)	
ESPECIFICACIONES	
Essentials	
Estado	Launched
Número de procesador	Q8400
Cantidad de núcleos	4
Cantidad de subprocesos	4
Velocidad del reloj	2.66 GHz
Caché L2	4 MB
Relación bus/núcleo	8
Paridad FSB	No
Conjunto de instrucciones	64-bit
Advanced Technologies	
Tecnología de virtualización Intel® (VT-x)	 Yes
Intel® 64	 Yes
Estados de inactividad	Yes
Tecnología Intel SpeedStep® mejorada	 Yes
Tecnologías de monitoreo térmico	Yes
Bit de desactivación de ejecución	Yes

Figura E.2 Características procesador Intel Q8400

ANEXO F: COSTOS DE EQUIPOS UTILIZADOS EN LA IMPLEMENTACIÓN DEL PROTOTIPO



Discomse
DISTRIBUIDOR DE COMPUTADORES & SERVICIOS INFORMATICOS
discomse1@yahoo.com - discomse@andinet.net

GUANIN ALOMOTO MELBA GEORGINA
R.U.C. 1708684053001

AUTORIZACION SRI:
1106787985
MATRIZ Serie S002-001

FACTURA

0001446

PERSONA NATURAL OBLIGADA
A LLEVAR CONTABILIDAD

Saraguro E 7-295
y Taya
Telfs.: 2903-794
3080-421
09 9934-619
09 9919-370
Quito - Ecuador



Fecha de Emisión: Junio 9, 2009

Cliente: ESCUELA POLITECNICA NACIONAL
Ladrón de Guevara E11-253
Dirección:
R.U.C./C.I.: 1760005620001 Ciudad QUITO

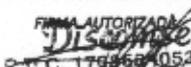
Forma de Pago: CONTADO
L.L.
Vendedor:
Telf.: 2507-190

CANTIDAD	DESCRIPCION	V. UNITARIO	V. TOTAL
3	COMPUTADORES MARCA : INTEL-CLONE ALTEK case minitwer ATX, fuente de 650 wats Mother Board INTEL DG-35EC PROCESADOR INTEL CORE 2 QUAD DE 2.6 GHz 4 MB Q 8400 FSB 1333 Mhz Sonido , Video y red (gigabit) incorporado SD/MMC/Memory Stick/CF/XD MEMORIA RAM DDR2 800 MHz 8 GB (4x2 Gb.) DOS DISCOS DUROS 500 GB SATA DVD WRITER 20X Teclado Multimedia PS/2, Mouse optico PS/2, parlantes GIGABYTE 10/100/1000 PCI Regulador de Voltaje	665,00	1995,00
3	Monitor Lcd de 22" L.G.	180,00	540,00
12	memorias para mainboard Intel D865PERL-DDR1	17,00	204,00
12	Memorias para Mainboard Intel D945PSN-DDR2	40,00	480,00
1	disco duro Externo USB 2.0 de 500 GB IOMEGA	100,00	100,00
3	Discos duros SATA 1 TB (internos) MAXTOR	118,00	354,00
1	NAS (network Attached Storage) de 2 bahias	292,00	292,00
2	Discos duros SATA 1 TB (compatible con el NAS) MAXTOR	118,00	236,00

CARGUENAS BENEFIZ FRI FICCO TELEFONO: HUC 1738438708001 autorizacion 1946, 13 MAYUZO 2006, 98 1401 @ 1800

Debo y pagare incondicionalmente a la orden de DISCOMSE, en el lugar que me convenga, el valor total expresado en este documento, más los impuestos legales respectivos y el interés máximo legal por mora, autorizado por la Junta Monetaria para Bancos e Instituciones Financieras, más todos los gastos que ocasione su cobro, siendo, suficiente prueba para ellos, la mera conversación del acreedor. Sin pretexto, estimes de presentación para el pago, así como de aviso por falta de este hecho, renuncia domicilio y me someto a los jueces o tribunales de Quito y al trámite Ejecutivo o Verbal Sumario a elección de DISCOMSE cada y tratara en cualquier momento los derechos que ampara el presente documento sin que sea necesaria notificación alguna ni nueva aceptación de mi parte.

RECIBI CONFORME 

FIRMA AUTORIZADA

R.U.C. 1708684053001
Firma Autorizada

Subtotal 4.201,00

IVA 0%

Descuento

0% % I.V.A.

TOTAL 4.201,00

Nombre: _____

COMPUTADORES - NOTEBOOKS - REDES - SUMINISTROS - SERVICIO TECNICO

					Av. 10 de Agosto N29-140 y Cuero Caicedo, Edificio: Vivanco, Ofic. Telf: (593) (2) 2501027 - 2560319		
PROPUESTA ECONOMICA							
Cliente: Ana Guerrero				Fecha: Quito, Octubre 19 del 2010			
Dirección:				No. Cotización: 087			
Teléfono: 095-057942				EJECUTIVO: ECH			
Ciudad: Quito							
RUC:							
Atención: Ana Guerrero							
ITEM	CANT	MARCA	MODELO	DESCRIPCION	P. UNIT. USD	P. TOTAL USD	OBSERVACIONES
Opción 1:							
1	1		DFE-520	Computador: Procesador Dual Core	\$ 498.00	\$ 498.00	NUEVO
2	1	CNET	PCI	Switch CGS-800 8 puertos	\$ 71.50	\$ 71.50	NUEVO
3	1	D-Link	DFE-520	KVM Switch 4 Puertos	\$ 69.99	\$ 69.99	NUEVO
4	6		PCI	Cable UPT cat 5	\$ 0.60	\$ 6.00	NUEVO
SUB-TOTAL						\$ 645.49	
SUBTOTAL						\$ 645.49	
12% (IVA)						\$ 77.46	
TOTAL						\$ 722.95	
SON:							
CONDICIONES GENERALES							
VALIDEZ DE LA OFERTA:		15 DIAS O HASTA AGOTAR STOCK					
FORMA DE PAGO:		50% ANTICIPO, 50% CONTRA ENTREGA					
TIEMPO DE ENTREGA:		2 DIAS EXEPTO EN MODULOS DE FIBRA Y EQUIPOS CISCO QUE ES 4 SEMANAS					
GARANTÍA:		DE FABRICA					
IMPUESTO A APLICAR		12% IVA					
IMPORTANTE							
Equipos entregados localmente							
Disponemos de servicios de instalación, mantenimiento y soporte técnico a pedido del cliente							
Precio exclusivo y confidencial para el cliente							
Los precios que se presentan responden únicamente a los precios que aparecen en la cotización							
Este número de oferta reemplaza a cualquier oferta emitida anteriormente							
TECNIT Ing. Eduardo Chuquizán		Aprobado por: Cliente Ana Guerrero		Firma Cliente			
					Fecha:		

ANEXO G: HERRAMIENTA DE GESTIÓN *RIGHTSCALE*

RightScale es una plataforma de gestión para administrar infraestructura de *Cloud Computing* de múltiples proveedores. En este proyecto se utilizó la herramienta de gestión *Elasticfox* por ser gratuita; *RightScale* es una herramienta comercial, sin embargo se pudo acceder a una cuenta limitada gratuita para mostrar que existen otras opciones para gestionar una Nube. Otra opción también es *LandScape* una herramienta de Canonical⁸⁶.

Para acceder se puede utilizar el enlace ubicado en la interfaz web de UEC con credenciales del administrador (Figura G.1). También se muestran los parámetros requeridos para el registro de la Nube con *RightScale* (Cloud URL y Cloud ID).

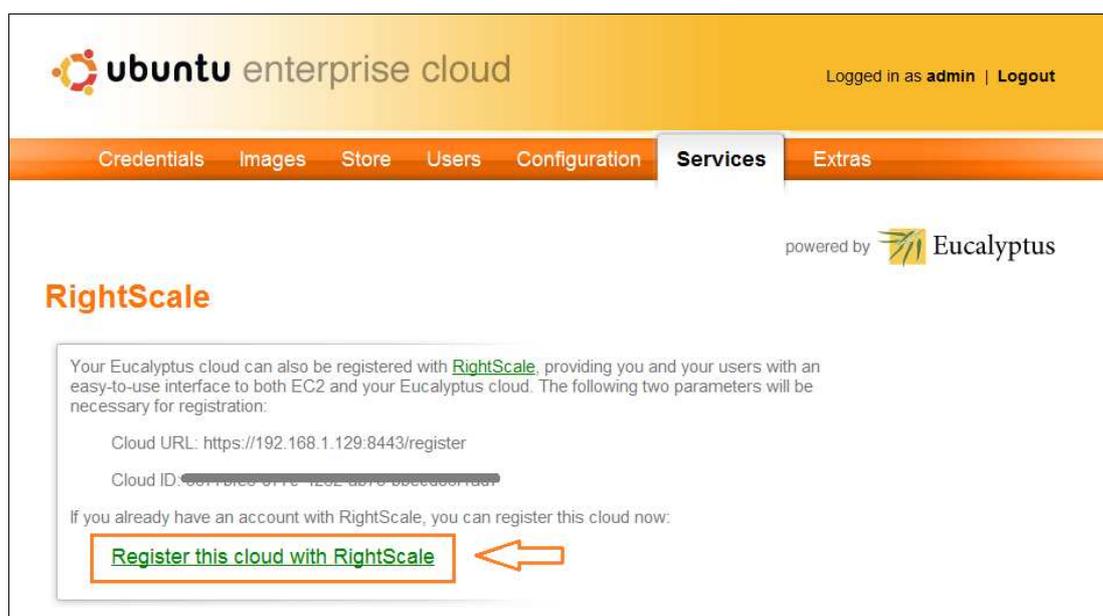


Figura G.1 Enlace a *RightScale* desde UEC

Para ingresar se debe tener una cuenta y se requiere una dirección de correo electrónico (Figura G.2) a través de la cual se envían notificaciones.

⁸⁶ Más información en: <http://www.canonical.com/enterprise-services/ubuntu-advantage/landscape>.

Figura G.2 Página de ingreso a *RightScale*

Una vez que se ha ingresado es necesario registrar la Nube con los siguientes parámetros (Figura G.3), los cuales generalmente se establecen por defecto siempre y cuando se utilice el link ubicado en la Figura G.1.

Figura G.3 Registro de la Nube con *RightScale*

Como se puede observar en la Figura G.3 se requiere que esté abierto el puerto 8443 para acceder a la Nube, cabe destacar que como *RightScale* está en Internet el *front-end* requiere tener una IP pública.

Finalmente se puede observar las mismas funciones que se pueden realizar a través de *Elasticfox* (Figura G.4), esta vez desde cualquier navegador web, en este caso se utilizó Google Chrome.



Figura G.4 Nube de *Eucalyptus* registrada con *RightScale*

ANEXO H: HERRAMIENTA DE MONITOREO MRTG

Se ha utilizado la herramienta MRTG (*Multi Router Traffic Grapher*) que no es un gestor de red, pero permite visualizar de forma gráfica, a través de una interfaz web, la carga de tráfico de interfaces de red específicas. Recoge datos de dispositivos que manejan SNMP⁸⁷ (*Simple Network Management Protocol*), es por eso que el dispositivo que se quiera monitorizar debe tener instalado, configurado y en ejecución, el agente SNMP.

Una vez instalados los paquetes necesarios para realizar el monitoreo, al configurar SNMP se debe establecer el nivel de seguridad con el que se trabajará en SNMP a través de la autenticación, en este caso se está utilizando un nombre de comunidad SNMP del tipo pública (*public*), se transforma el nombre de seguridad (*readonly* y *local*) a un nombre de grupo y cada grupo tiene determinados permisos de acceso (Código H.1), finalmente solo se debe reiniciar el servicio.

```
com2sec readonly default      public
com2sec local default        public

group MyROGroup v1          readonly
group MyROGroup v2c         readonly

group MyRWGroup v1          local
group MyRWGroup v2c         local

view all    included .1
view system included .iso.org.dod.internet.mgmt.mib-2.system

access MyROGroup "" any noauth exact all none none
access MyRWGroup "" any noauth exact all all all

syslocation Unknown (configure /etc/snmp/snmpd.local.conf)
syscontact Root <root@localhost> (configure /etc/snmp/snmpd.local.conf)
```

Código H.1 Archivo de configuración snmpd.conf

⁸⁷ **SNMP** es el protocolo básico para la administración de la red.

Por otro lado, al configurar MRTG en `mrtg.cfg`, es necesario manejar ciertos parámetros generales como:

- RunAsDaemon: yes
- EnableIPv6: no
- WorkDir: /var/www/mrtg
- Options[_]: bits,growright
- WriteExpires: Yes

A continuación, al utilizar `cfgmaker`, se añade al archivo de configuración, información sobre los dispositivos que manejen SNMP, en este caso se ha monitoreado la red local, es decir, el tráfico de las interfaces pública (`eth1`) y privada (`eth0`) del *front-end*, que ya se analizaron anteriormente con la herramienta *iptraf* en la Sección 3.6.

Finalmente, se debe crear la página web que contenga la información de las interfaces de red monitoreadas.

A continuación se muestra el resultado del monitoreo de las interfaces de red `eth0` y `eth1` del *front-end*. La Figura H.1 muestra el tráfico que atraviesa la interfaz `eth0`, los resultados de esta gráfica se presentan diariamente cada 5 minutos, la Figura H.2 presenta el mismo tráfico anterior pero con resultados obtenidos cada 30 minutos y de forma semanal, la Figura H.3 permite visualizar el tráfico de la interfaz `eth1` diariamente y cada 5 minutos, finalmente en la Figura H.4 se puede observar el tráfico de la Figura H.3 pero semanalmente y cada 30 minutos.

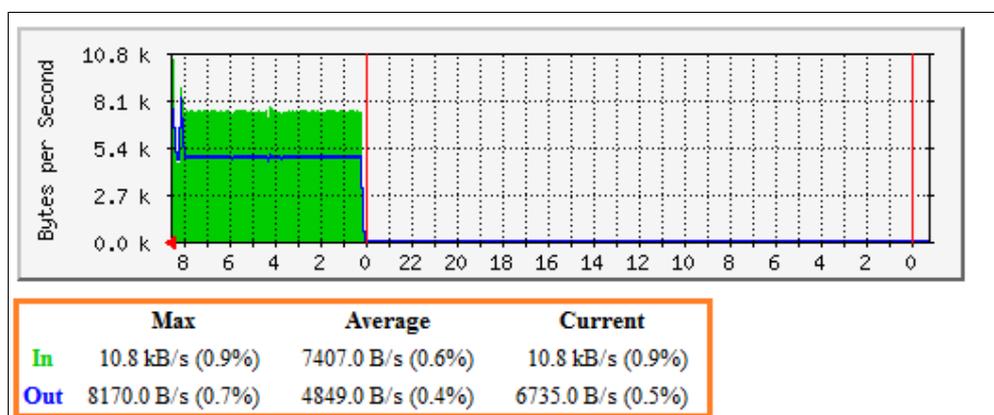


Figura H. 1 Tráfico en eth0 cada 5 minutos

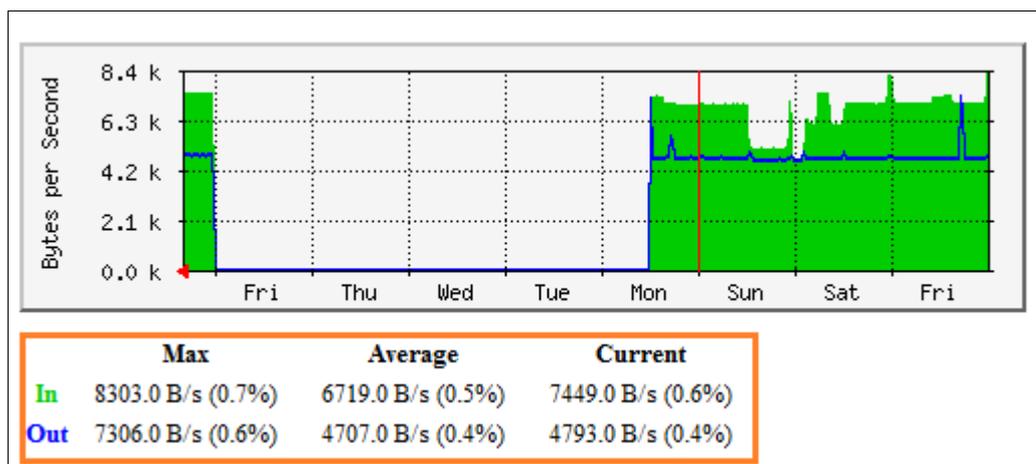


Figura H.2 Tráfico en eth0 cada 30 minutos

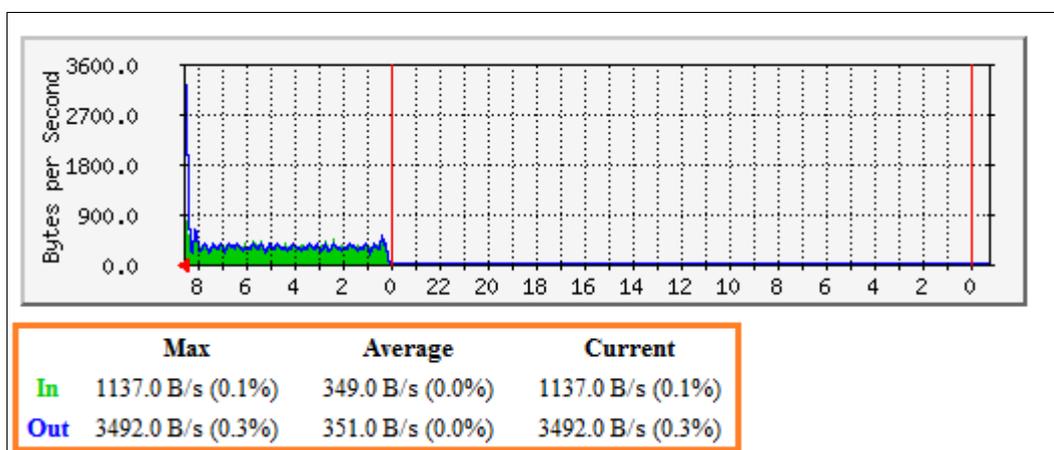


Figura H.3 Tráfico en eth1 cada 5 minutos

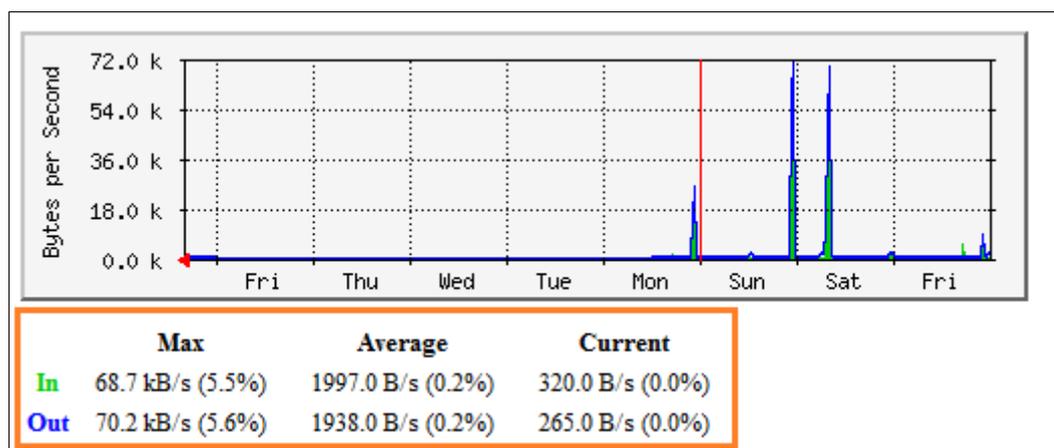


Figura H.4 Tráfico en eth1 cada 30 minutos

Con las gráficas anteriores, se puede visualizar el tráfico entrante (*In*) y el tráfico saliente (*Out*) de las interfaces de red. Se puede determinar fácilmente que el

tráfico total que atraviesa la interfaz eth0 es superior en comparación a los resultados del tráfico de la interfaz eth1, ya que la interfaz de red eth0 se comunica con los nodos que son los que ejecutan las MVs en el prototipo.

ANEXO I: MANUAL DE CONFIGURACIÓN PARA USUARIOS NO ADMINISTRADORES

Aplicar para una cuenta de usuario en la dirección:

<http://cloud.epn.edu.ec/> en el enlace “login”. (Figura I.1)

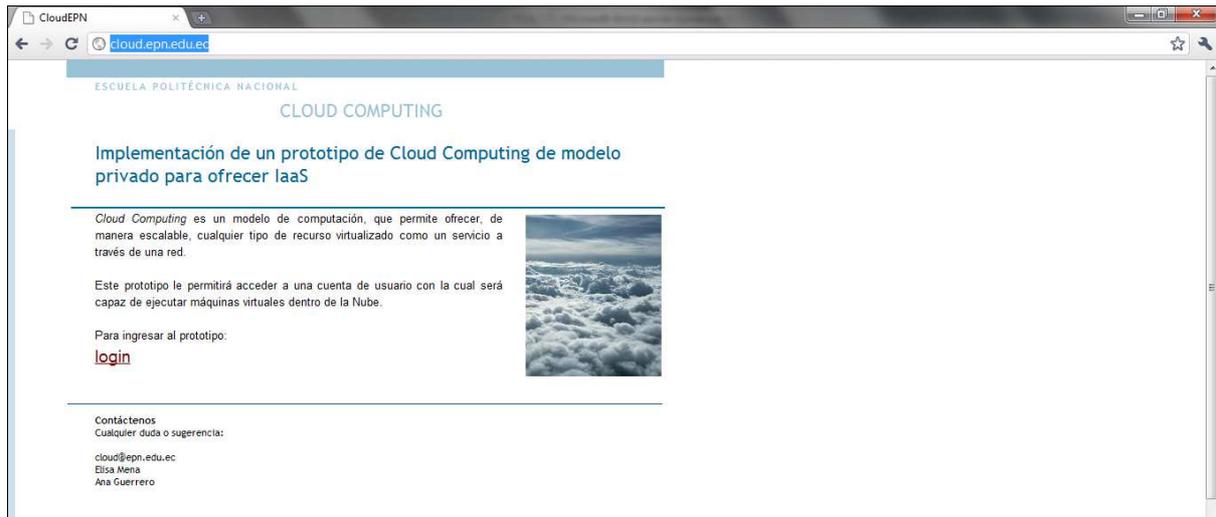


Figura I.1 Ingreso a la Nube SmartCloud

En la interfaz de login, ingresar a “Apply”. (Figura I.2)

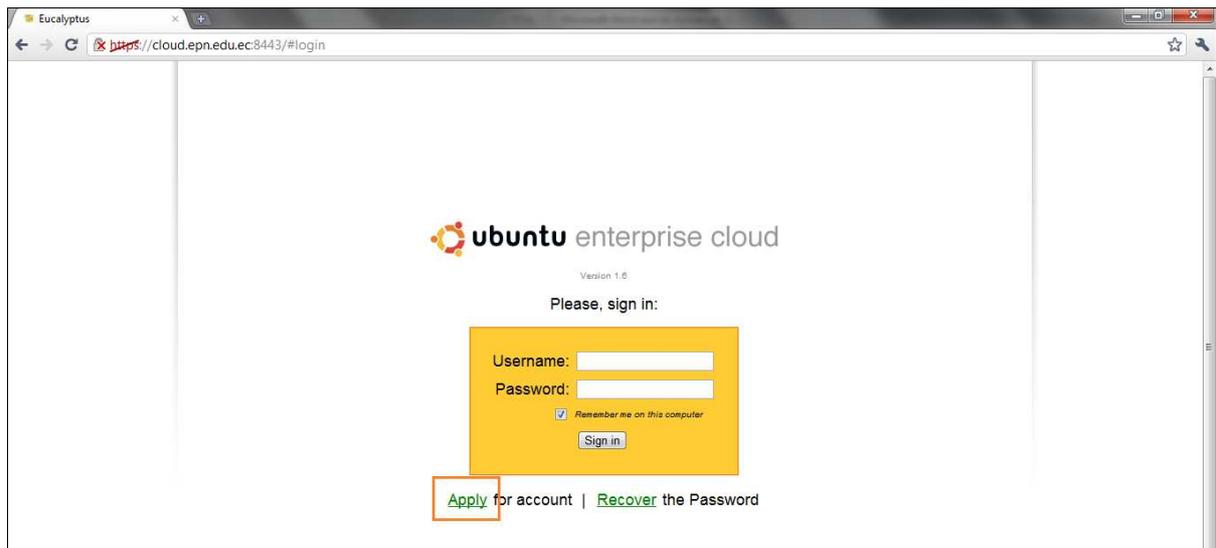


Figura I.2 Aplicación a una nueva cuenta

En el formulario de aplicación llenar los campos requeridos. En el campo correo electrónico (*Email address*, Figura I.3) llenar con una cuenta de correo del dominio @cloud.epn.edu.ec previamente solicitada al Administrador.

Figura I.3 Aplicación a una nueva cuenta

Una vez que la cuenta ha sido creada, revisar el correo electrónico (Figura I.4) ya que es necesario dar clic en el enlace enviado para poder activar la cuenta.

Figura I.4 Cuenta de correo electrónico del Nuevo usuario

El enlace se muestra en el Figura I.5.

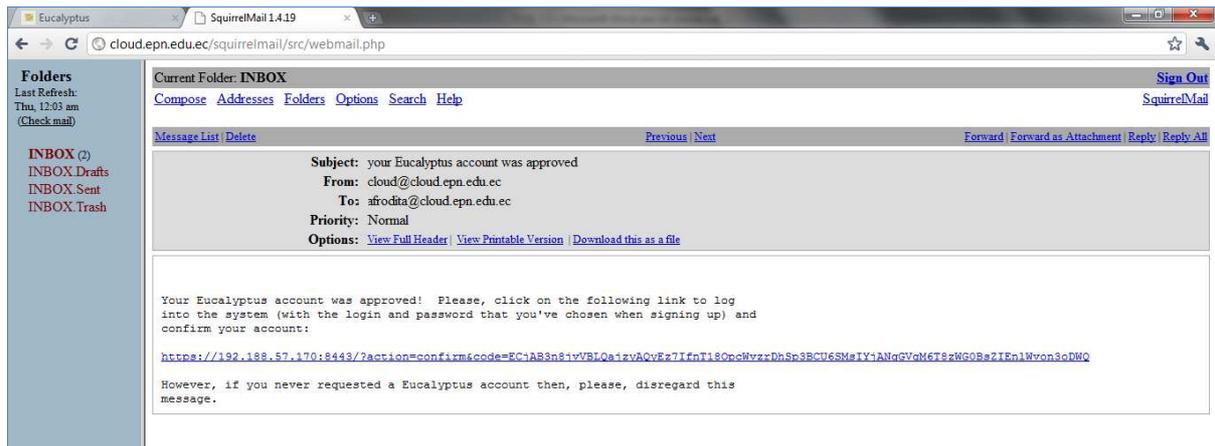


Figura I.5 Enlace de activación de cuenta

Una vez activada la cuenta se puede ingresar al sistema (Figura I.6) a través de la interfaz de Login (Figura I.2).

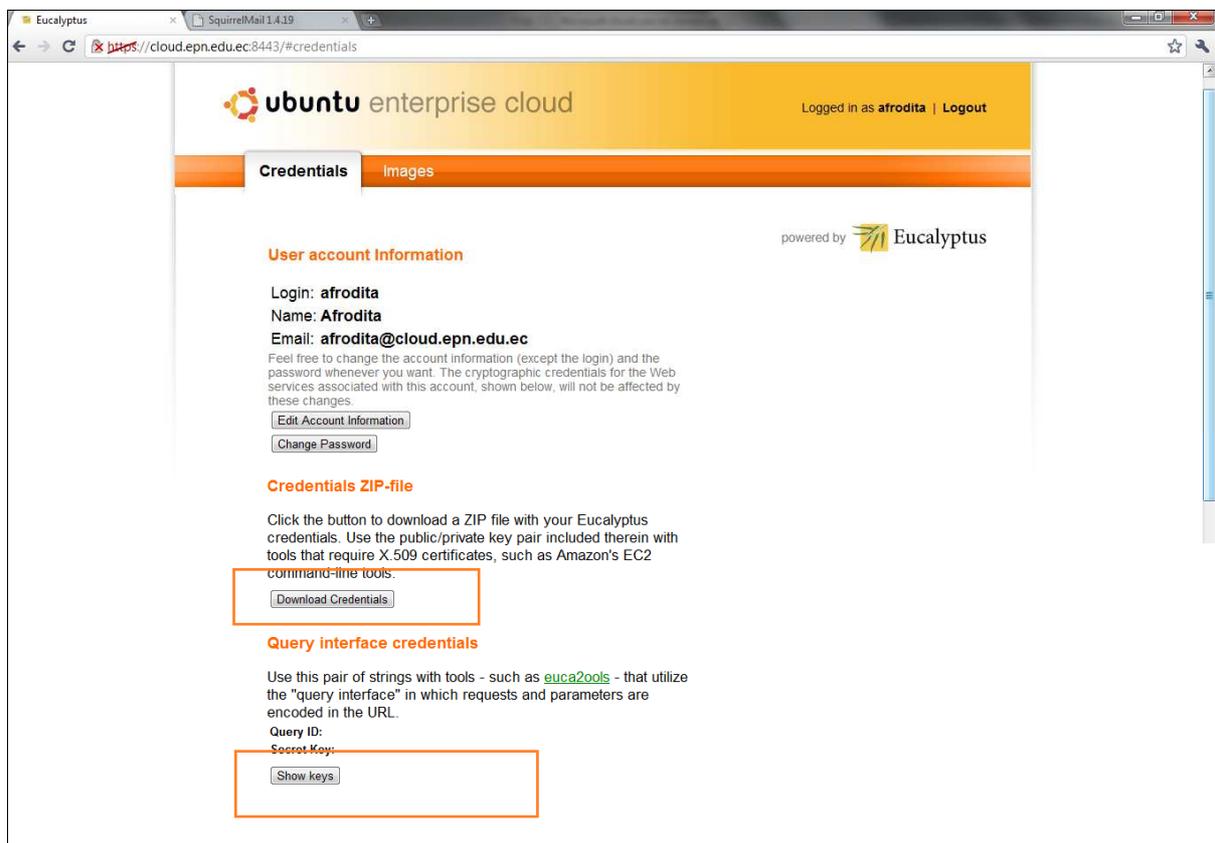


Figura I.6 Cuenta del Nuevo usuario en UEC

La Figura I.7 muestra las credenciales (Credentials ZIP-file) y los strings (Query Interface credentials) necesarios para conectarse a una Nube.

Descargar (Download Credentials) las credenciales y almacenarlas en un lugar seguro.

INSTALACIÓN Y CONFIGURACIÓN DE HYBRIDFOX

Descargar hybridfox

<http://code.google.com/p/hybridfox/downloads/detail?name=hybridfox-1.6.000019.xpi&can=2&q=>

y abrir el archivo .xpi con el Navegador web Mozilla Firefox.

En la pestaña *Tools* de Firefox escoger *Hybridfox* (Figura I.7).

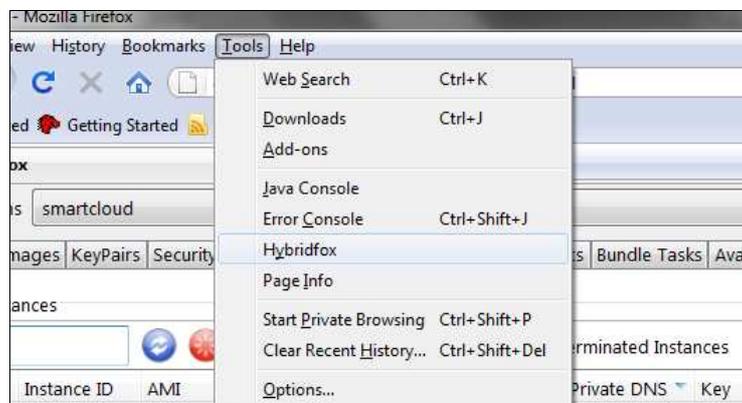


Figura I.7 Hybridfox en Firefox

Luego configurar una Region llamada *smartcloud* en el botón **Regions** de *Hybridfox* (Figura I.8) con Endpoint URL: <http://cloud.epn.edu.ec:8773/services/Eucalyptus>

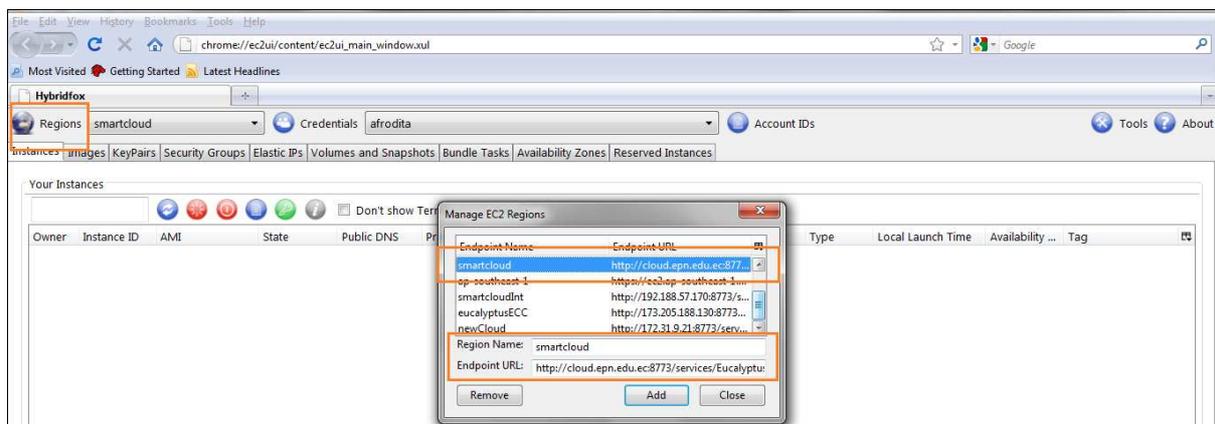
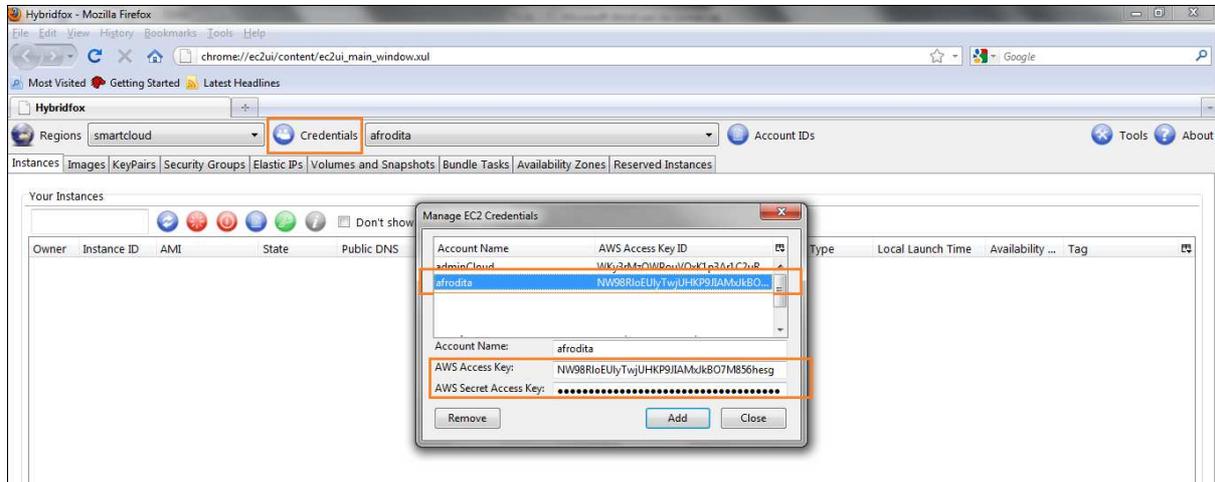


Figura I.8 Región smartcloud

Luego es necesario configurar las credenciales del usuario en el botón *Credentials* de *Hybridfox*. AWS Access Key es Query ID de la Figura I.6, I.9 y AWS Secret Access Key es Secret Key de la Figura I.6, I.9.



Query interface credentials

Use this pair of strings with tools - such as [euca2ools](#) - that utilize the "query interface" in which requests and parameters are encoded in the URL.

Query ID: NW98RloEUlyTjwJHkP9JIAMxJkBO7M856hesg
Secret Key: ~~NW98RloEUlyTjwJHkP9JIAMxJkBO7M856hesg~~

Figura I.9 Configuración de Credenciales

Hybridfox ya ha sido completamente configurado. Después se debe crear un par de llaves (*keypairs*, Figura I.10). Se les debe asignar un nombre.

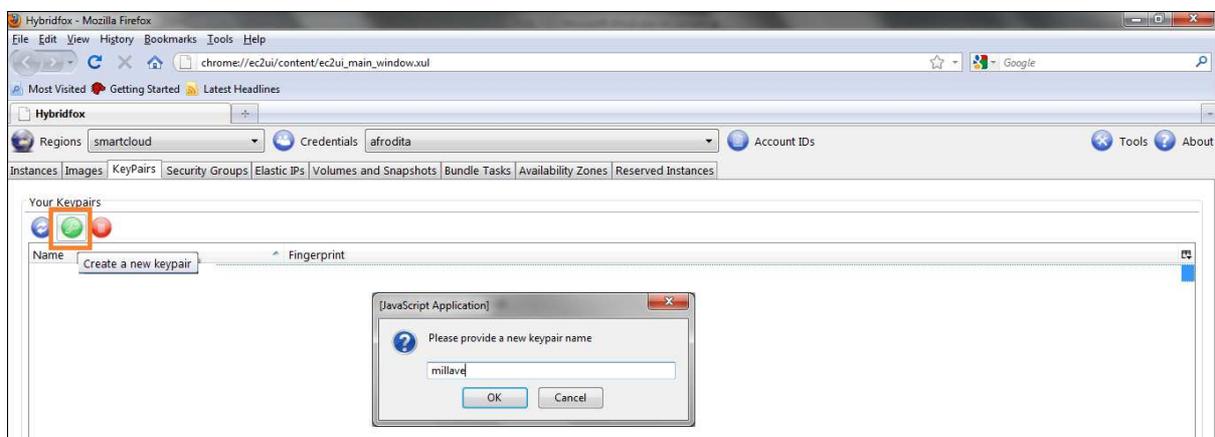


Figura I.10 Creación de *keypairs*

La llave pública se almacena en *Eucalyptus* y la otra (privada) en algún lugar de la máquina del usuario.

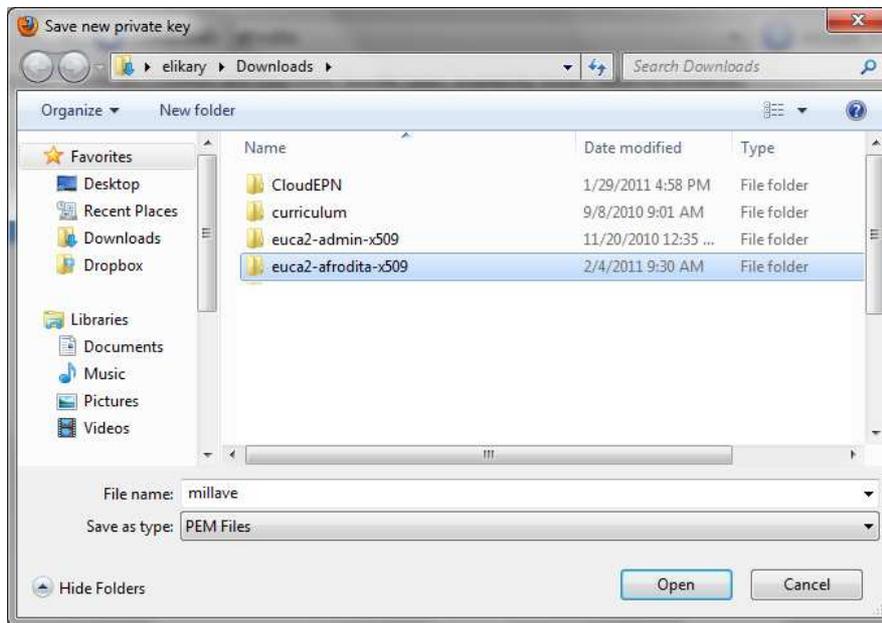


Figura I.11 Llave privada *keypairs*

Con las llaves creadas es posible ejecutar y acceder a una máquina virtual, para lo cual se puede escoger de la pestaña *Images* el identificador emi de una de las imágenes disponibles en el sistema y dar clic en el botón *Launch Instances* de color verde (Figura I.12).

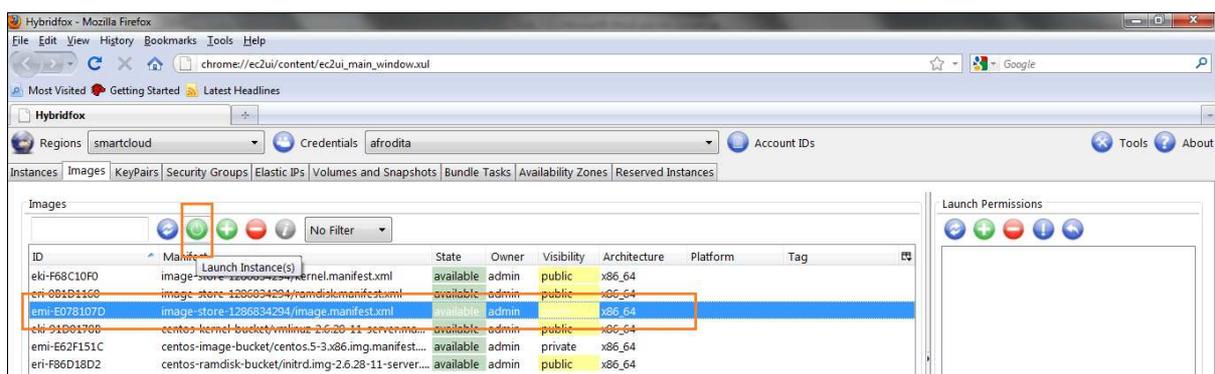


Figura I.12 Ejecutar una instancia

Aparecerá un formulario para ejecutar la máquina virtual, en donde es necesario llenar algunos campos de acuerdo a los requerimientos.

Instance type: el tamaño de la máquina virtual. Number of instances: Número de máquinas virtuales que se quieren ejecutar con la imagen escogida. Public Address: si se desea asignar una dirección IP pública (del pool de IPs públicas disponibles) a la interfaz pública de la máquina virtual. Launch in: Grupo de seguridad en el que va a estar la máquina virtual, todos los usuarios tienen por defecto el grupo *default* (Figura I.13).

The screenshot shows the 'Launch new instance(s)' dialog box with the following configuration:

- AMI ID: emi-E078107D
- AMI Tag: (empty)
- AMI Manifest: image.manifest.xml
- AKI ID: eki-F68C10F0
- ARI ID: eri-0B1D1160
- Instance Type: c1.medium
- Minimum number of instances: 1
- Maximum number of instances: 1
- New Instance(s) Tag: (empty)
- KeyPair: millave
- Availability Zone: <any>
- Additional Info: (empty)
- Public Address:
- Security Groups: (empty)
- Available Groups: (empty)
- Launch in: default
- User Data: (empty)
- Buttons: Open File, Open Binary File, Launch, Cancel

Figura I.13 Ejecución de una máquina virtual

Se puede ver la máquina virtual ejecutándose con los datos que se configuraron previamente (Figura I.14).

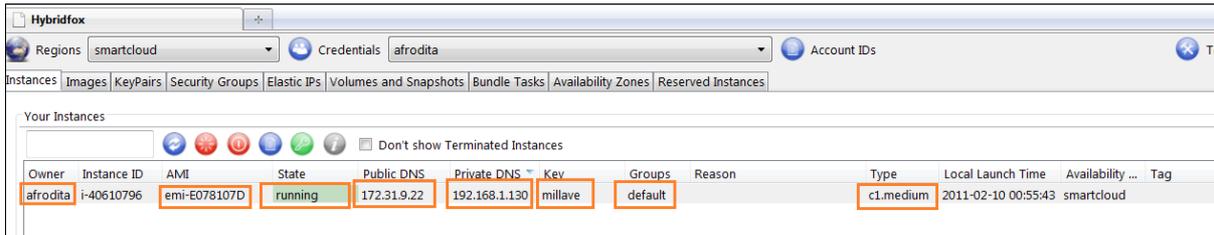


Figura I.14 Máquina virtual ejecutándose

Para conectarse a la nueva máquina virtual se requiere configurar el cliente SSH en *Hybridfox* en el botón *Tools* (Figura I.15).

SSH Comand: el path de ubicación del ejecutable de *Putty*, SSH user: el usuario con el que se conecta a la máquina virtual en este caso “ubuntu” ya que la imagen usada fue la de Ubuntu, si hubiera sido Centos el usuario por defecto es “root”.

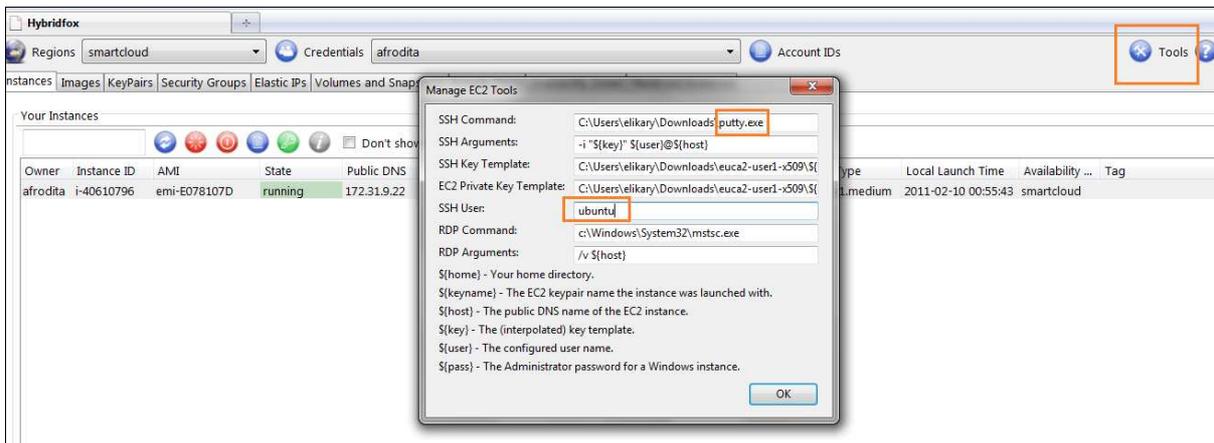


Figura I.15 Configuración del cliente SSH

Para conectarse a la instancia se puede hacer este procedimiento en *Hybridfox* con el botón verde con forma de llave, después pedirá la llave privada del par de las *keypairs* generadas (Figura I.10), esta tiene extensión *.pem*, pero *putty* trabaja con extensiones *.ppk*, así que a través de *puttygen* se puede hacer la transformación necesaria.

Se carga el archivo *.pem* en *puttygen* y después se lo guarda con el *nombre.ppk* (Figura I.16, Figura I.17).

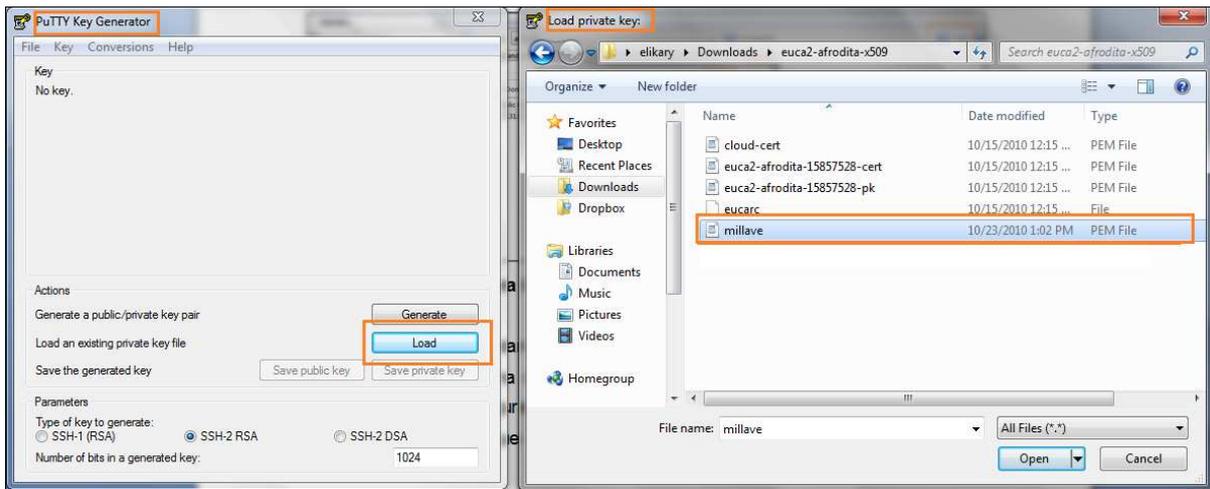
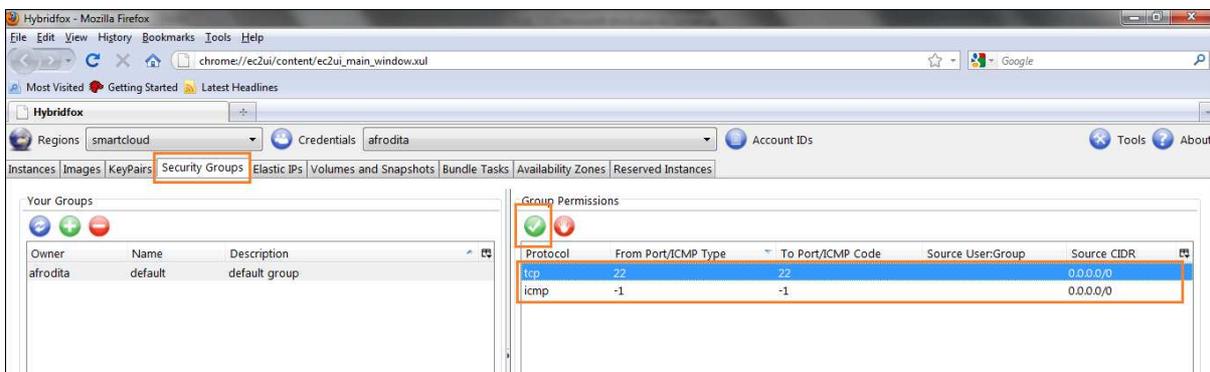
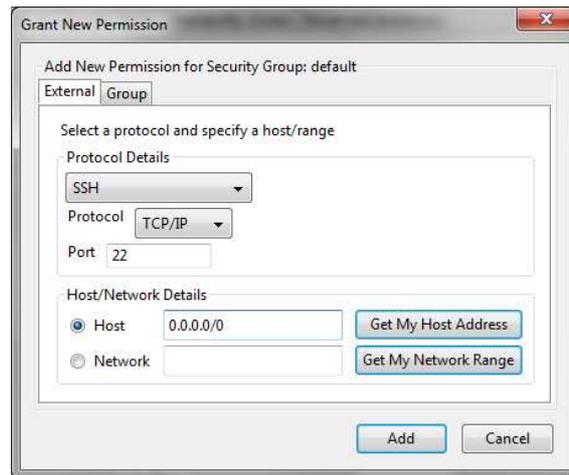


Figura I.16 Transformación .pem a .ppk

Después es necesario configurar el grupo de seguridad para que la máquina virtual acepte tráfico de ICMP y SSH, esto se configura en la ventana *Grant New Permission* (Figura I.17b) que se muestra desde el botón verde con un visto (Figura I.17a).



(a)



(b)

Figura I.17 Security group default afrodit

Así es posible conectarse a la instancia a través de SSH con la llave .ppk generada (Figura I.18).

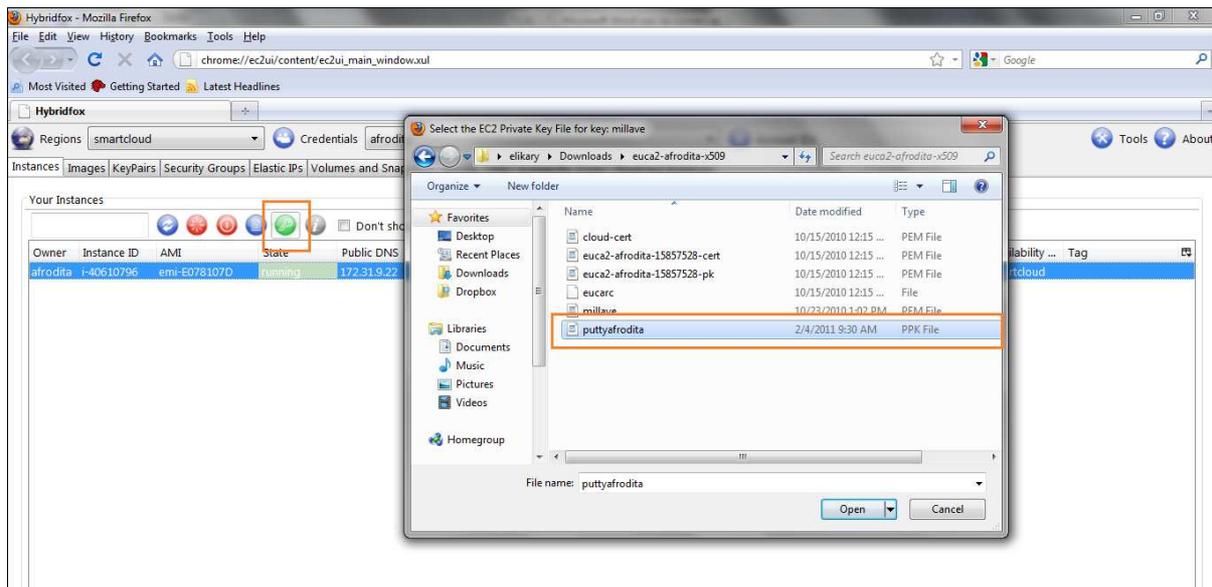


Figura I.18 Llave .ppk

Y después se inicia sesión en la instancia. (figura I.19)

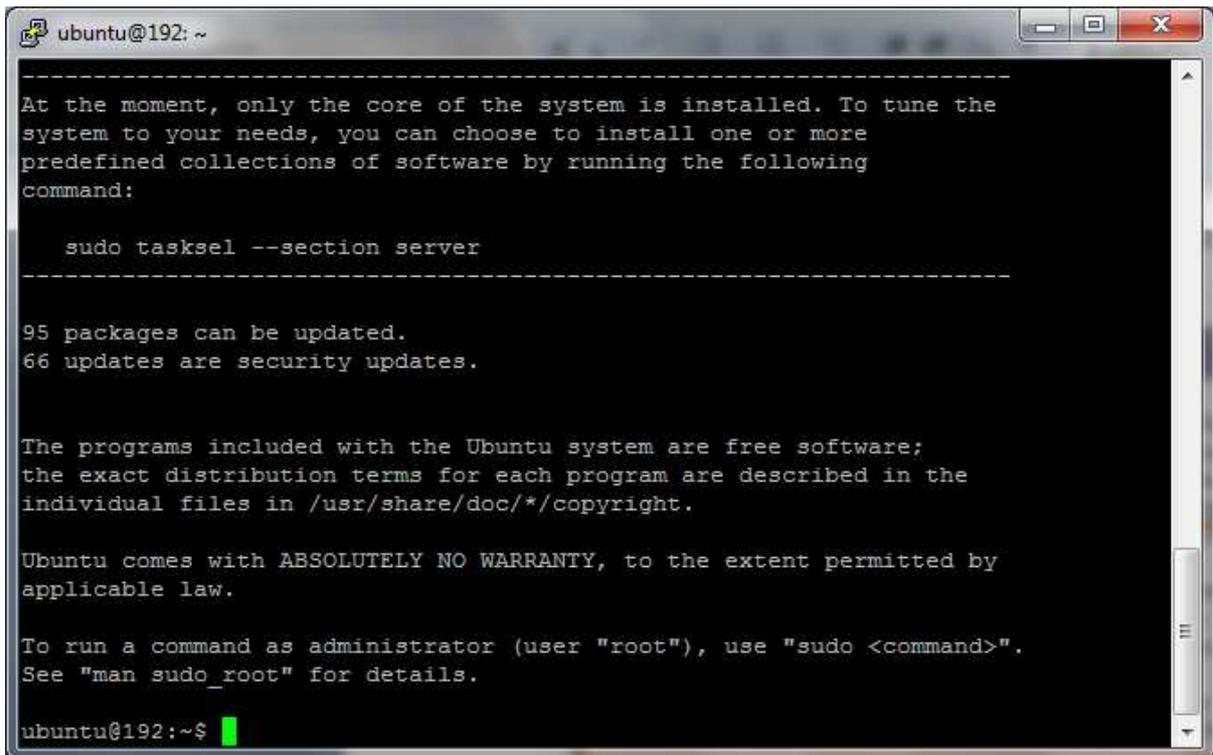


Figura I.19 Sesión en una instancia

Se puede crear un volumen para adjuntarlo a una máquina virtual (Figura I.19). En la pestaña *Volumes and Snapshots* de *Hybridfox*, se define el tamaño en Gigabytes y un nombre vol1 (Figura I.20).

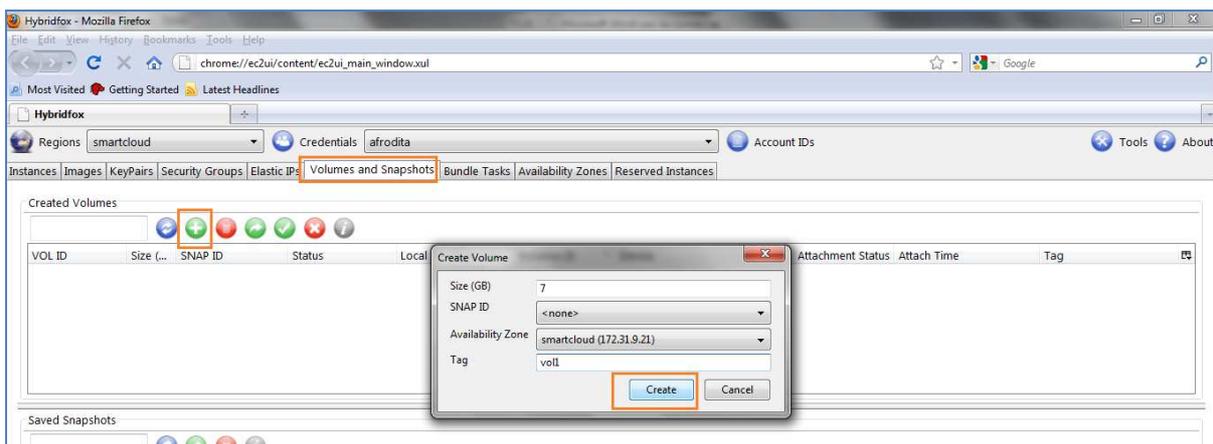


Figura I.20 Creación de un volumen lógico

Después se adjunta el volumen a una máquina virtual, se coloca el id de la máquina virtual y el dispositivo que reconocerá el sistema operativo (Figura I.21).

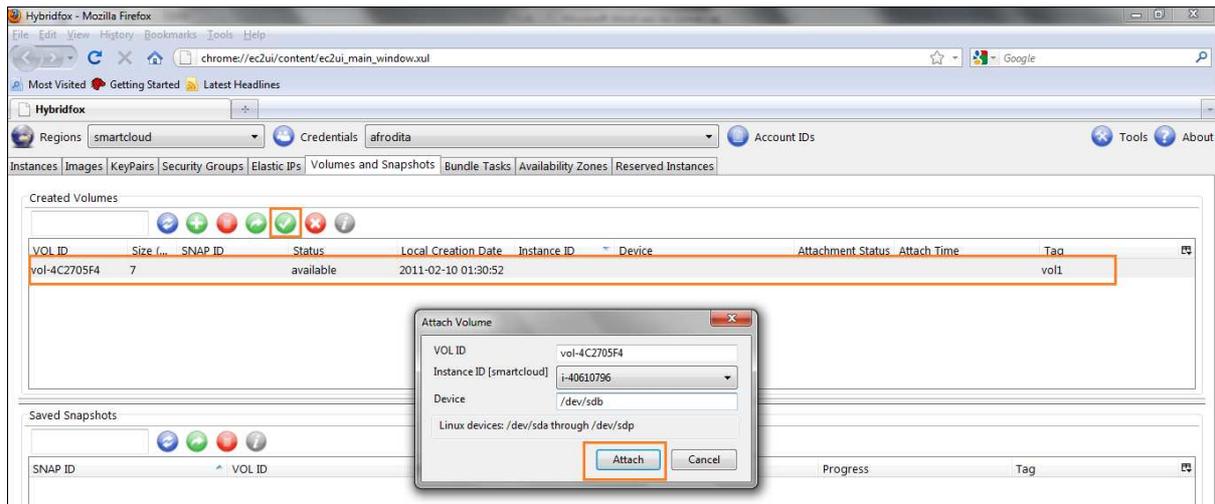


Figura I.21 Adjuntar un volumen a una máquina virtual

Con el comando `fdisk -l` se puede comprobar el dispositivo reconocido por el sistema operativo (Figura I.22)

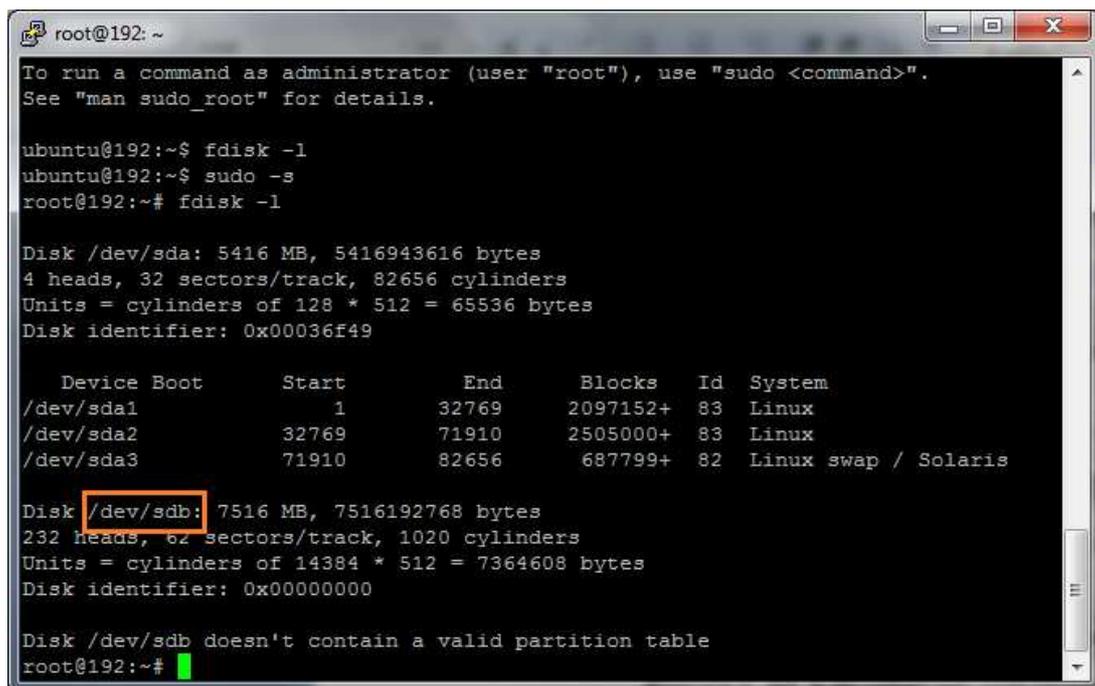


Figura I. 22 Volumen reconocido como un dispositivo

Para hacer uso del volumen es necesario darle formato, y luego montarlo en un determinado punto de montaje (Figura I.23).

```

root@192: ~
232 heads, 62 sectors/track, 1020 cylinders
Units = cylinders of 14384 * 512 = 7364608 bytes
Disk identifier: 0x00000000

Disk /dev/sdb doesn't contain a valid partition table
root@192:~# mkfs.ext3 /dev/sdb
mke2fs 1.41.9 (22-Aug-2009)
/dev/sdb is entire device, not just one partition!
Proceed anyway? (y,n) y
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
458752 inodes, 1835008 blocks
91750 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1879048192
56 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Writing inode tables: 7/56

```

Figura I.23 Formato a un volumen lógico

Se crea el punto de montaje y se monta el volumen (Figura I.24) y ya se puede hacer uso de éste dispositivo.

```

root@192: ~
Proceed anyway? (y,n) y
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
458752 inodes, 1835008 blocks
91750 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1879048192
56 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 28 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
root@192:~# mkdir /volumen
root@192:~# mount /dev/sdb /volumen/
root@192:~#

```

Figura I.24 Montaje del volumen lógico

Se puede crear un *snapshot* de un volumen con el botón señalado en la Figura I.25 *Create new snapshot from this volume* y contar con la misma información que se haya guardado en el volumen original.

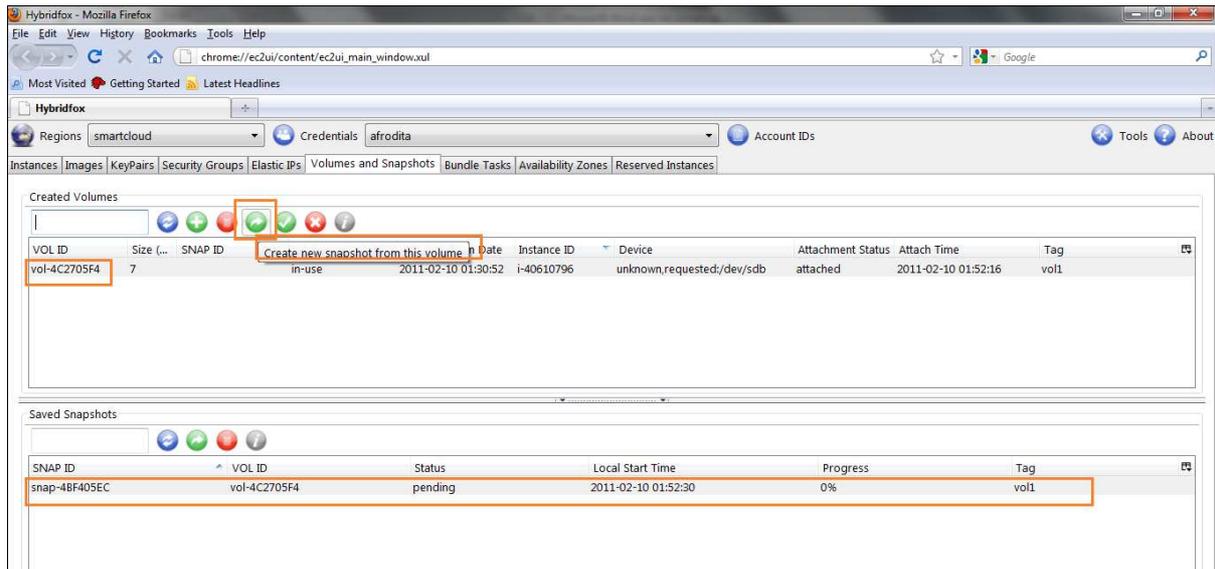


Figura I. 25 Crear un snapshot de un volumen

Y finalmente se puede crear un nuevo volumen a partir del *snapshot* del volumen original (Figura I.26).

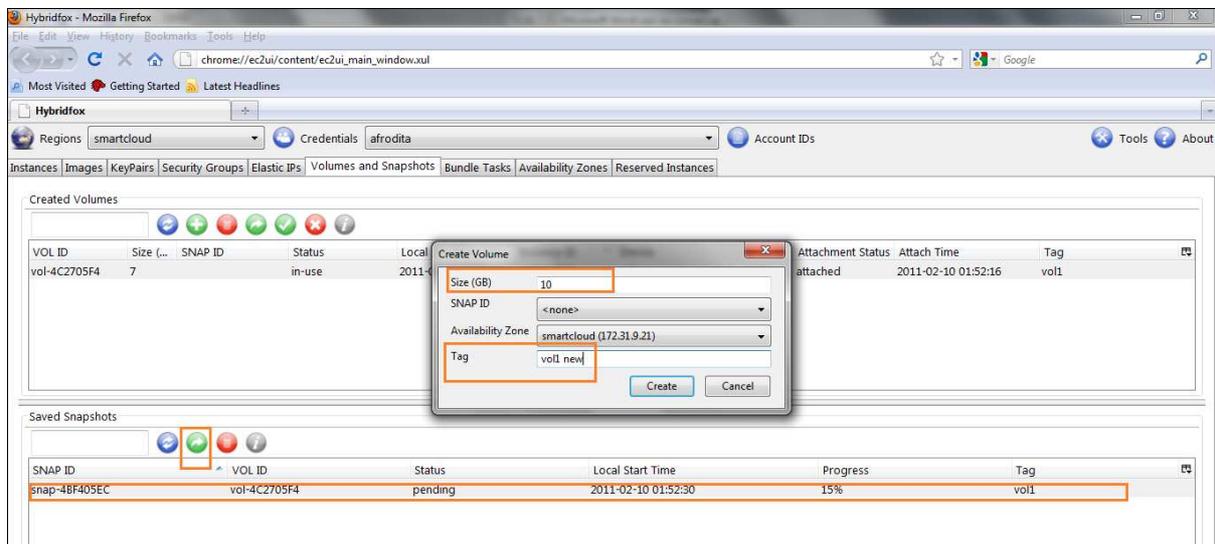


Figura I.26 Volumen desde un *snapshot*