

FACULDADE ANTONIO MENEGHETTI
JOSÉ LUIZ DA SILVA RODRIGUES FILHO

**IMPLANTAÇÃO DE UMA INFRAESTRUTURA DE COMPUTAÇÃO EM NUVEM
IAAS COM OPENSTACK**

RECANTO MAESTRO
2014

JOSÉ LUIZ DA SILVA RODRIGUES FILHO

**IMPLANTAÇÃO DE UMA INFRAESTRUTURA DE COMPUTAÇÃO EM NUVEM
IAAS COM OPENSTACK**

Trabalho de Conclusão de Curso-Monografia,
apresentado como requisito parcial para obtenção do
título de Bacharel em Sistemas de Informação, Curso de
Graduação em Sistemas de Informação, Faculdade
Antonio Meneghetti - AMF.

Orientador: Prof. Ms. Daniel de Moraes João

RECANTO MAESTRO
2014

JOSÉ LUIZ DA SILVA RORIGUES FILHO

**IMPLANTAÇÃO DE UMA INFRAESTRUTURA DE COMPUTAÇÃO EM NUVEM
IAAS COM OPENSTACK**

Trabalho de Conclusão de Curso-Monografia,
apresentado como requisito parcial para obtenção do
título de Bacharel em Sistemas de Informação, Curso de
Graduação em Administração, Faculdade Antonio
Meneghetti - AMF.

Orientador: Prof. Ms. Daniel de Moraes João

COMISSÃO EXAMINADORA

Prof. Ms. Daniel de Moraes João
Orientador do Trabalho de Conclusão de Curso
Faculdade Antonio Meneghetti

Profa. Dra. Ana Bulegon
Membro da Banca Examinadora
Faculdade Antonio Meneghetti

Prof. Ms. Fábio Prass
Membro da Banca Examinadora
Faculdade Antonio Meneghetti

Recanto Maestro, 06 de dezembro de 2014.

AGRADECIMENTOS

Gostaria de dar início aos meus agradecimentos, principalmente agradecendo à formação que recebi na Antonio Meneghetti Faculdade. Posso dizer que foi uma feliz coincidência estudar aqui e analisando o contexto, o tempo que passou e todas as oportunidades, confirmo, foi muito positivo para minha vida e formação contínua como humano.

Os semestres passaram rapidamente, mas os ensinamentos compartilhados – teóricos e práticos – ficarão para a vida toda, assim como a inspiração e a coragem, obrigado!

Ao meu orientador, professor Daniel João, pela dedicação em compartilhar sua capacidade de relação e companheirismo ao encarar um trabalho sobre o qual deixaria de ser iniciante para ser parte. Ao professor Gustavo Réquia, pela sua dedicação e coragem ao me dar tanto incentivo à forma abstrata de como se apresentavam os temas em minha mente e a sempre me fazer buscar mais, nunca deixando que me fizesse contente, e me colocando em uma permanente busca pelos meus objetivos acadêmicos. Com toda a certeza este trabalho só foi possível ser desenvolvido com a persistência e a coragem que vocês apostaram a cada frase relacionada ao tema computação em nuvem.

Nos nomes de meu irmão Rodrigo, minha irmã Lenise, minha mãe Luiza, meu pai José Luiz, meu tio Itamar, minha tia Dulce e a Gabrielle agradeço a todos os meus familiares pelo apoio e principalmente porque sempre me deram o espaço necessário para que eu me tornasse cada vez mais eu, para que minha identidade fosse cada vez mais minha, por compreender e, principalmente, respeitar todas as minhas decisões, sem nunca deixar a sinceridade e o apoio incondicional. Obrigado!

Aos meus colegas de aula, companheiros de curso, amigos! Vocês me ensinaram tanto sobre a vida, quanto sobre as relações, aprendi que o companheirismo se constrói tijolo por tijolo, dia após dia, dificuldade após dificuldade, com vocês dividi momentos que jamais deixarão de existir no infinito de minha memória e de minha história, obrigado!

Àqueles que fiz a opção de chamar de amigos, no sentido mais puro da palavra, meu muito obrigado. O apoio quase que incessante, os e-mails trocados, as chamadas de atenção, tudo foi muito válido e esse trabalho existe também porque

vocês se dedicaram a me ouvir e ajudar. Aelton, Éber, Vicente, Norman, Magnum, Diego, Renan Matheus e Dieimes um muito obrigado!

“Cada sonho que você deixa para trás, é um pedaço de seu futuro que deixa de existir”.

Steve Jobs

RESUMO

O objetivo deste trabalho é desenvolver uma forma de implementar uma infraestrutura de computação em nuvem escalável para ambientes que necessitem de *IaaS*. Para este tipo de demanda ou processo, as empresas costumam basear-se em metodologias bem definidas e mundialmente disseminadas como os processos ITIL. Este trabalho apresenta uma solução através da ferramenta Openstack, e demonstra que através do método proposto na pesquisa reduz-se o tempo de provisionamento de recursos como o armazenamento, conectividade, plataformas, aplicações e serviços, escalável e de alta disponibilidade, utilizando as inovações no modelo de computação em nuvem na maneira de se entregar serviços de computação em nuvem, para auxiliar as equipes de implantação de serviços de TI. No final é disponibilizada uma imagem virtual (*appliance*) da instalação do Openstack.

Palavras-chave: Computação em Nuvem; Openstack; Linux; Ubuntu; Processos.

ABSTRACT

The objective of this work is to develop a way to implement a scalable cloud computing infrastructure for environments that require IaaS. For this type of demand or process, companies often based on well-defined methodologies and disseminated worldwide as the ITIL processes. This paper presents a solution by OpenStack tool, and shows that by the method proposed in the search reduces the resource provisioning time as the storage, connectivity, platforms, applications and services, scalable and high availability by using innovations in cloud computing model in the way they deliver cloud computing services to assist the teams deployment of IT services. At the end is provided a virtual image (appliance) installing OpenStack.

Keywords: Cloud Computing; Openstack; Linux; Ubuntu; Processes.

LISTA DE FIGURAS

Figura 1 - Comparação de pesquisa dos termos Servidor Dell, Servidor IBM, Servidor HP e IaaS.....	14
Figura 2 - Comparação de pesquisa dos termos OpenStack, CloudStack e Amazon AWS.....	15
Figura 3 - Diagrama pilha de computação em nuvem	18
Figura 4 - Arquitetura da Computação em Nuvem (VECCHIOLA; CHU; BUYYA, 2009)	21
Figura 5 - Diagrama da arquitetura de virtualização	25
Figura 6 - Diagrama do Software OpenStack.....	28
Figura 7 - Arquitetura de Implantação do sistema OpenStack.....	30
Figura 8 – Primeiro Login Openstack	42
Figura 9 – XenCenter: Tela de Login.....	43
Figura 10 – Tela de Login Openstack.....	43
Figura 11 – Tela de Dashboard Horizon do Openstack	44
Figura 12 - Configurar Segurança do <i>Openstack</i>	45
Figura 13 – Geração de Chaves de Segurança para Instâncias do Openstack	46
Figura 14 – Tela de Criação de uma Instância.....	47
Figura 15 – Tela de Seleção de Chave de Segurança para uma Instância	48
Figura 16 – Tela de Inicialização de uma Instância	49
Figura 17 - Infraestrutura	51
Figura 18 - Escalabilidade.....	53
Figura 19 - Gerenciamento	54
Figura 20 - Custo.....	55

Sumário

1	INTRODUÇÃO	10
1.1	PROBLEMA DE PESQUISA	12
1.2	OBJETIVOS	12
1.2.1	Geral	12
1.2.2	Específicos	13
1.3	JUSTIFICATIVA	13
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	COMPUTAÇÃO EM NUVEM	16
2.1.1	IaaS	19
2.1.2	PaaS	19
2.1.3	SaaS	19
2.1.4	Arquitetura da Computação em Nuvem	20
2.1.5	Modelos de Implantação	22
2.2	VIRTUALIZAÇÃO	22
2.2.1	Hypervisor (VMM)	24
2.2.2	XenServer	24
2.2.3	Citrix XenCenter	26
2.3	OPENSTACK	26
2.3.1	Arquitetura do Openstack	30
2.4	NAS	31
2.4.1	Freenas	31
2.5	LINUX	32
2.6.1	MISSÃO	33
2.7	UBUNTU LINUX	33
3	METODOLOGIA	35
3.1	DELINEAMENTO DA PESQUISA	35
3.2	HARDWARE	36
3.2	INSTALAÇÃO DE PACOTES XENAPI	37
3.3	INSTALAÇÃO DO DEVSTACK	39
3.3.1	Requisitos Esquisitos	39
3.3.2	Processo Macro de Instalação	39
3.3.3	ROTEIRO DE INSTALAÇÃO (install_os_domU.sh)	40
3.3.4	Etapa 1: Instalar XenServer	40
3.3.5	Etapa 2: Baixar o devstack	41
3.3.6	Etapa 3: Configurar o arquivo "localrc" localizado dentro do diretório "devstack"	41
3.3.7	Etapa 4: Executar o script ./install_os_domU.sh a partir do diretório "tools/xen"	42
4	RESULTADOS E DISCUSSÃO	50
4.1	RESULTADO COMPARATIVO ENTRE O MÉTODO TRADICIONAL E O PROPOSTO SOBRE ENTREGA DE SERVIÇOS DE INFRAESTRUTURA DE TI...	50
4.2	SEPARAÇÃO DA INTELIGÊNCIA DA ARQUITETURA E DA IMPLANTAÇÃO	56
4.3	RESULTADOS DA PRODUÇÃO DE UMA MÁQUINA VIRTUAL (APPLIANCE) COM OPENSTACK	57
5	CONSIDERAÇÕES FINAIS	59
	REFERÊNCIAS	61

1 INTRODUÇÃO

As mudanças geradas pela passagem do tempo modificam todas as áreas do conhecimento, seja pelo aperfeiçoamento desse conhecimento, seja por sua modificação ou surgimento de novas ideias que se agreguem a ele. A área de Tecnologia da Informação (TI), que em sua definição já abriga vertentes, se expande em todas as direções. Organizar e classificar informações para que elas se transformem em utilidade, em equipamentos que facilitem os processos profissionais são as ações que, por sua simplicidade e complexidade, ao mesmo tempo, garantem o poder àquele que melhor manipular essa totalidade.

Facilmente percebe-se como a área de TI tem sofrido grandes transformações nos últimos anos, além de um constante crescimento e expansão. Hoje, crianças habitam-se aos dispositivos móveis mais facilmente do que se habitam a ir à escola, por exemplo. A velocidade de acesso às informações é cada vez mais rápida e nos incomoda-se profundamente quando essa diminui ou, pior ainda, a conexão “cai”.

Nesse cenário, tem-se pessoas acessando cada vez mais a internet e querendo a disponibilidade imediata de informações que julgam interessantes – textos, fotos, músicas, documentos etc. Porém, esse volume crescente e contínuo de informação precisa ser armazenado e processado em algum momento em algum lugar. Para atender toda esta demanda, que é latente, fornecedores de TI criaram plataformas, adquiriram equipamentos e desenvolveram softwares com diferentes formas de acesso. Porém isso elevou de forma exponencial o custo para manter disponíveis todas essas informações.

Alguns problemas provenientes de equipamentos, atualizações constantes de hardware, consumo de energia, aumento dos custos de manutenção de equipamentos e infraestrutura, entre outros, têm feito com que os fornecedores e profissionais de TI busquem novas formas e métodos mais eficientes visando otimizar o aproveitamento dos recursos computacionais.

Os processos de TI também mudaram. Atualmente para disponibilizar um recurso ou serviço para um cliente é necessário arquitetar a solução com um especialista, viabilizar hardware e alocar uma equipe para execução do projeto. Isto

eleva o tempo dos projetos, visto que não há uma forma de se acompanhar as altas demandas do mercado com processos tecnologicamente defasados e lentos.

Esta realidade potencializa o *Cloud Computing* ou Computação em Nuvem, que pode ser considerada uma nova arquitetura de TI, uma evolução natural dos modelos baseados em *mainframe* e cliente/servidor. Conforme Velte, Velte e Elsenpeter (2009, p. 3), a função da computação em nuvem é “cortar custos operacionais e, o mais importante, permitir que departamentos de TI se concentrem em projetos estratégicos em vez de manter o data center funcionando”. Ou seja, com o modelo *Cloud Computing* muda-se a forma de adquirir a TI comercialmente, pois paga-se pelo uso dos recursos que se adaptam à demanda, reduzindo os custos operacionais. Este fenômeno também é conhecido como elasticidade. Além disso desonera o departamento de TI de preocupar-se com o ambiente físico dos dados (data center), permitindo direcionar o seu foco a projetos maiores e alinhados cada vez mais com o negócio.

Um dos modelos de se entregar computação em nuvem é a *Infrastructure as a Service* (IaaS), traduzida como Infraestrutura como um Serviço. É a forma de entregar processamento, armazenamento, rede e energia como um serviço, permitindo expansão/redução conforme a demanda, sem a necessidade do cliente adquirir todo o ambiente físico para tal função.

Há uma série de iniciativas que trabalham com este modelo de elasticidade, contudo a mais recente e bem-sucedida é o *Openstack*, uma plataforma de código aberto de computação em nuvem.

O *Openstack* contempla praticamente todo o ambiente de uma infraestrutura que é ofertada como serviço IaaS, ou seja, ele pode ser entendido como um grande sistema operacional na nuvem do modelo *Cloud Computing*, onde, por meio de várias APIs (*Application Programming Interface*) desenvolvidas, é possível orquestrar (gerenciar) todos os recursos virtualizados por ele.

Com base nisto, percebe-se a importância do assunto, bem como as características técnicas que esta nova forma de adquirir serviços pode fornecer, como por exemplo, elasticidade, otimização de recursos de hardware, organização de data centers e orquestração de serviços. Ou seja, as empresas investem seus recursos no uso e não na aquisição de aplicativos e equipamentos de TI.

Recentemente os usuários vem cada vez mais conhecendo formas detalhadas deste novo formato de aquisição de TI. Isto se deve principalmente ao aumento de

empresas que fornecem infraestruturas, plataformas e softwares como um serviço utilizando como meio a internet.

1.1 PROBLEMA DE PESQUISA

Tendo em vista o panorama inicial, o problema desta pesquisa é identificar como os fornecedores e profissionais de TI podem entregar soluções de processamento, armazenamento de dados, conectividade, plataformas, aplicações e serviços de forma segura, escalável e de alta disponibilidade aos seus clientes.

A razão desta pesquisa é solucionar dois problemas. Primeiro, o tempo de implantação (entrega) de um sistema operacional para os clientes, tomando-se como base desde a arquitetura da solução até o provisionamento de um equipamento hardware. Para este tipo de demanda ou processo, as empresas costumam basear-se em metodologias bem definidas e mundialmente disseminadas como os processos ITIL (Information Technology Infrastructure Librabry).

O segundo problema é automatizar o processo de implantação (método) da infraestrutura, onde após a implantação deste método será gerado um arquivo de instalação com a automatização do processo de implantação da arquitetura de computação em nuvem baseada em *Infrastructure as a Service* ou Infraestrutura como um Serviço. Este processo diminuiria drasticamente o tempo de implantação desta arquitetura, permitindo assim também separar a inteligência da arquitetura da implantação.

Um dos principais problemas de se trabalhar com tecnologias de alto nível é que muitas vezes é necessário o melhor técnico e/ou o melhor projetista para realizar a implantação. Neste modelo pode-se separar o arquiteto da solução do responsável técnico pela implantação, reduzindo também o nível de investimentos em formação e qualificação direta para todos os envolvidos e, sim, focando em suas respectivas áreas de atuação.

1.2 OBJETIVOS

1.2.1 Geral

Para solucionar os problemas citados foram efetuados estudos de uma nova forma de fornecer tecnologia, denominada computação em nuvem, que tem como principal propósito disponibilizar poder de processamento, armazenamento de dados, conectividade, plataformas, aplicações e serviços de forma segura, escalável e de alta disponibilidade.

Portanto a proposta para a solução desta problemática será “desenvolver uma forma de implementar uma infraestrutura de computação em nuvem escalável para ambientes que necessitem de *IaaS*”.

1.2.2 Específicos

a) Demonstrar que através deste método proposto reduz-se o tempo de provisionamento de recursos como o armazenamento, conectividade, plataformas, aplicações e serviços, escalável e de alta disponibilidade.

b) Comparar o tempo de uma implantação com e sem este método.

c) Separar o conceito de arquiteto de soluções de TI do responsável pela instalação.

d) Produzir uma máquina virtual (*appliance*) de instalação de um Sistema Operacional de base Linux para a fácil e rápida instalação de uma infraestrutura de computação em nuvem.

1.3 JUSTIFICATIVA

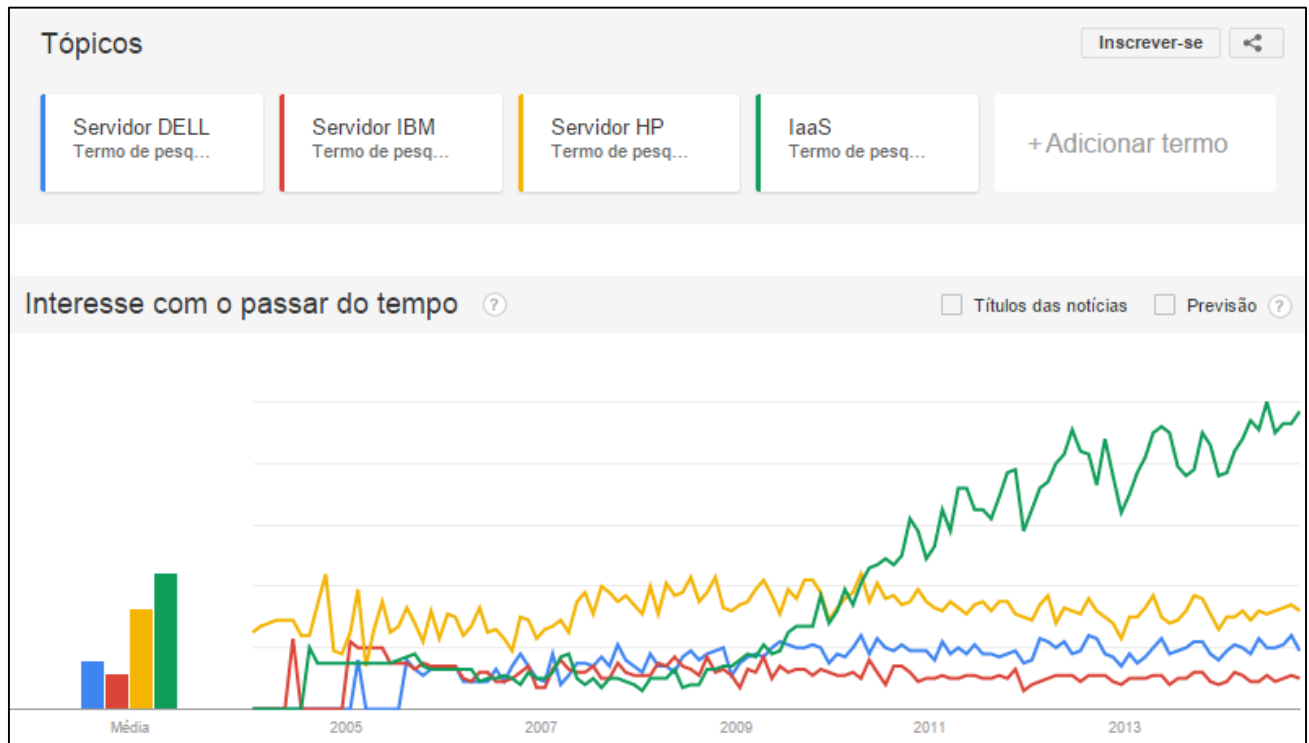
A escolha do tema se deu pela familiaridade e qualificação que o autor deste trabalho tem com infraestruturas de TI, principalmente quando se fala em sistemas operacionais Linux, virtualização, alta disponibilidade e processos de TI.

Além disso, o desenvolvimento deste trabalho irá fornecer subsídios técnicos e científicos para a eficiência na arquitetura e na entrega de serviços de computação em nuvem, enquanto benefício da empresa do autor.

Como benefício profissional, gera um conhecimento aprofundado de novos processos e tecnologias, além de direcionar o autor para uma pesquisa de mestrado e doutorado em um leque de possibilidades como segurança, alta disponibilidade, virtualização, processos de TI e governança de TI.

Como benefício tecnológico, justifica-se a escolha do tema pela tendência de crescimento do IaaS, como pode-se observar nas pesquisas do termo na figura 1:

Figura 1 - Comparação de pesquisa dos termos Servidor Dell, Servidor IBM, Servidor HP e IaaS



Fonte: <http://www.google.com.br/trends>

O Google disponibiliza uma ferramenta de comparação entre termos pesquisados chamada Google Trend. Na figura acima observa-se a comparação entre os seguintes termos pesquisados: 'Servidor DELL', 'Servidor IBM', 'Servidor HP' e 'IaaS'. Como pode-se observar na figura 1 em meados de 2010 o termo 'IaaS' (cor verde) disparou nas pesquisas do Google, destacando-se e superando os demais termos citados.

Há uma série de ferramentas disponíveis para gerenciamento de ambientes de computação em nuvem, contudo a ferramenta selecionada para este trabalho é a ferramenta chamada Openstack.

Dentre as soluções para gerenciamento de infraestruturas de computação em nuvem destaca-se o Openstack que é relativamente um recém-chegado no espaço da IaaS, tendo seu primeiro release lançado no final de 2010. Apesar da presumida falta de maturidade da solução, e dado que ela existe há menos de cinco anos, o

Openstack atualmente é uma das pilhas de nuvem mais utilizada no mundo. No entanto, em vez de ser uma solução individual, o Openstack é um conjunto crescente de soluções de código aberto incluindo projetos principais e recentemente desenvolvidos que, juntos, formam uma pilha de IaaS eficiente e madura (IBM, 2014).

Pesquisando na ferramenta Google Trend os termos: 'Openstack', 'Cloudstack' e 'Amazon AWS', que são ferramentas de gerenciamento de IaaS, pode-se observar na Figura 2 que em meados de 2012 as pesquisas pelo termo 'Openstack' superaram os demais termos pesquisados, destacando-se assim no mercado:

Figura 2 - Comparação de pesquisa dos termos openstack, Cloudstack e Amazon AWS



Fonte: <http://www.google.com.br/trends>

No próximo capítulo será abordada a fundamentação teórica para complementar e aprofundar os termos supracitados ampliando assim, o conhecimento para dar base ao tema proposto.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 COMPUTAÇÃO EM NUVEM

A Computação em Nuvem pode ser entendida como uma tecnologia que permite novos modelos de negócios, onde os recursos computacionais como processamento, armazenamento, rede e software, são ofertados através da internet, permitindo assim que sejam acessados de forma remota (ARMBRUST et al., 2009). Todas as informações ficam constantemente disponíveis aos usuários e eles podem acessá-las de qualquer lugar e a qualquer momento. Cada componente de uma nuvem computacional é ofertado como um serviço e são normalmente alocados em data centers, utilizando assim hardware compartilhado para computação, armazenamento e rede.

Os computadores de usuários dos serviços de Computação em Nuvem não necessitam de quaisquer configurações avançadas de hardware, diminuindo assim, o custo na aquisição e manutenção de recursos computacionais (SOUSA et al., 2009). A Computação em Nuvem surgiu e se tornou rapidamente uma das mais influentes tecnologias na indústria de TI e está revolucionando o modo como os recursos de TI são gerenciados e utilizados (ARMBRUST et al. 2009).

Um sistema pode ser considerado uma nuvem computacional se apresentar as características abaixo (MELL, 2011):

- a) Autoatendimento sob demanda: O consumidor deve ter a capacidade de provisionar recursos sem a necessidade de ação ou intervenção por parte do fornecedor.
- b) Acesso amplo a rede: Todos os recursos providos pela computação em nuvem devem ser acessíveis através da rede e em diferentes tipos de dispositivos móveis, computadores, etc.
- c) Agrupamento de recursos: Recursos de computação do provedor são agrupados para servir a vários consumidores com diferentes recursos físicos e virtuais dinamicamente atribuídos e de acordo com a demanda do consumidor. Há um senso de localização, em que o cliente geralmente não tem controle ou conhecimento sobre a exata localização geográfica dos recursos

disponibilizados, mas pode ser capaz de especificar o local mais próximo por exemplo: país, estado ou data center.

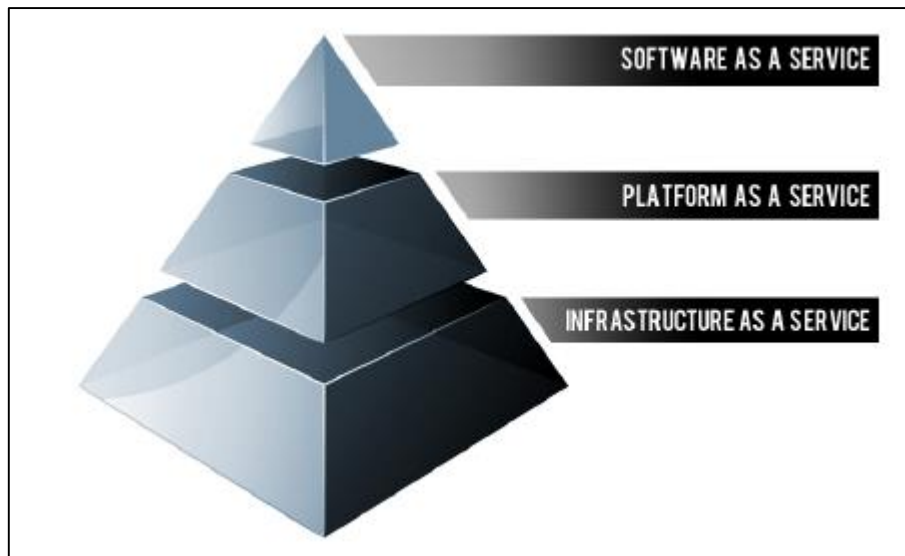
- d) Elasticidade: os recursos podem ser elasticamente infinitos do ponto de vista do usuário, pois o mesmo pode requisitar e liberar recursos a qualquer momento.
- e) Serviço mensurado: Os sistemas de nuvem podem controlar e monitorar os recursos utilizados. A mensuração para os serviços pode ser por exemplo: armazenamento, processamento, largura de banda e contas de usuários ativos. O uso de recursos pode ser monitorado, controlado e reportado, oferecendo assim um meio de cobrar dos consumidores os recursos computacionais em ambientes de computação em nuvem.

Cloud Computing é um paradigma de computação em larga escala que possui foco em proporcionar economia de escala, em que um conjunto abstrato, virtualizado, dinamicamente escalável de poder de processamento, armazenamento, plataformas e serviços são disponibilizados sob demanda para clientes externos através da Internet. (FOSTER, 2008, p. 1)

Tomando como base a definição de Foster (2008), pode-se observar que o foco é gerar economia de escala e oferecer serviços para clientes externos, o que pode ser percebido com força nos ambientes *Cloud*.

A figura 3 mostra a pilha de *Cloud Computing* - mostra três categorias distintas dentro do *Cloud Computing*: Software como um Serviço, Plataforma como um Serviço e Infraestrutura como um Serviço.

Figura 3 - Diagrama pilha de computação em nuvem



Fonte: <http://c179631.r31.cf0.rackcdn.com/cloudcomputestackimage1.png>

Na imagem acima, podemos ver as três categorias em detalhes e de forma muito simplificada para se diferenciar os modelos de *Cloud Computing*.

- a) Aplicações SaaS (Software as a Service) são projetadas para os usuários finais, entregues através da web;
- b) O PaaS (*Platform as a Service*) é o conjunto de ferramentas e serviços projetados para tornar a codificação (implantação) desses aplicativos rápida e eficiente; e
- c) IaaS é o hardware e software que opera tudo - servidores, armazenamento, redes e sistemas operacionais.

Para ajudar a entender como esses três componentes são relacionados, veremos algumas analogias:

Por si só, uma infraestrutura não é útil, ela fica lá esperando alguém para torná-la produtiva, resolvendo um problema particular. Imagine que o sistema de transporte interestadual no Brasil. Mesmo com todas essas estradas construídas, elas não seriam úteis sem carros e caminhões para o transporte de pessoas e bens. Nessa analogia, as estradas são a infraestrutura e os carros e caminhões são a plataforma que fica no topo da infraestrutura que transporta as pessoas e bens. Estas pessoas e bens podem ser considerados o software (Rackspace, 2013).

2.1.1 IaaS

Podemos definir que Infraestrutura como um Serviço (IaaS) é uma forma de entregar computação em nuvem, onde a infraestrutura de servidores, processamento, rede, armazenamento, e todo o ambiente necessário para o funcionamento são contratados como serviços. Ao invés de adquirir servidores, softwares, espaço em data centers, os clientes usam esses recursos como um serviço totalmente terceirizado sob demanda.

As ofertas de soluções, os custos, a alta disponibilidade e o nível de profissionalismo dos provedores de serviços *cloud* são os principais fatores da adoção do IaaS. As características principais do IaaS são: recursos são contratados como um serviço, podem ter um custo variável pelo uso ou definido de forma prévia, alta escalabilidade com rapidez e eficiência e monitoramento e gerenciamentos avançados (DUALTEC, [2012]).

2.1.2 PaaS

Podemos definir que Plataforma como um Serviço (PaaS) é uma maneira de entregar um ambiente de computação em camadas de soluções como serviço. As ofertas de PaaS facilitam a implantação de aplicações de menor custo e complexidade na compra e gestão do hardware, software e recursos de provisionamento de infraestrutura, que fornece todas as facilidades necessárias para suportar o ciclo de vida completo de construção e entrega de aplicações web e serviços totalmente disponíveis a partir da Internet.

O PaaS pode ser considerado um análogo ao SaaS, tornando-se diferente somente pelo fato de que ao invés de um software entregue pela web, é uma plataforma (um ambiente) para a criação, hospedagem e controle de software (DUALTEC, [2012]).

2.1.3 SaaS

O SaaS pode ser entendido como software sob demanda, é um modelo de entrega de software no qual o mesmo e seus dados associados são hospedados na

internet (nuvem) e normalmente acessados utilizando um navegador web através da internet.

Atualmente já utilizamos este conceito em praticamente todos os serviços de internet que consumimos, como um buscador na web (Google, Bing, Yahoo) ou até mesmo em seu webmail, Gmail e Outlook. De forma geral, o conceito de software como um Serviço é muito mais abrangente e mais aplicado no contexto de aplicações de negócios (DUALTEC, [2012]).

2.1.4 Arquitetura da Computação em Nuvem

A arquitetura de computação em nuvem é baseada em camadas, onde cada camada possui sua particularidade no fornecimento de recursos (BUYAYA et al., 2009).

A figura abaixo representa a arquitetura da computação em nuvem, assim como suas associações, onde é possível observar na figura que esta arquitetura possui quatro camadas que se deslocam de forma progressiva do sistema até o usuário final.

Figura 4 - Arquitetura da Computação em Nuvem (VECCHIOLA; CHU; BUYYA, 2009)



Fonte: http://www.teleco.com.br/tutoriais/tutorialservnuvopers1/clip_image006.gif

Observando de baixo para cima podemos observar que a primeira camada é a camada de infraestrutura física, que são os recursos de data center, onde normalmente contêm os dispositivos de armazenamento, servidores, e outros recursos de hardware.

Na camada de *middleware central* podemos observar que sua responsabilidade é gerenciar a infraestrutura física e fornecer um ambiente apropriado para as aplicações, assim como usufruir ao máximo dos recursos físicos. Podemos dividir esta camada em duas: uma que será responsável por manter o isolamento das aplicações na plataforma de hospedagem e outra responsável pelas intermediações para com os usuários como o gerenciamento de Qualidade de Serviço (QoS – *Quality of Service*) e Monitoramento.

Na camada de *middleware*, nível de usuário, temos a parte de programação da nuvem, ou seja, toda a plataforma de APIs, linguagens de programação e bibliotecas, todos os recursos necessários para o desenvolvimento de aplicações, esta camada não é de interesse de usuários finais, normalmente é destinada a usuários mais experientes como programadores e engenheiros de software.

Por fim a camada de aplicação que destina-se aos usuários finais, a camada de aplicação pode ser entendida como a única camada de interesse dos usuários finais, pois os mesmos acessam as aplicações como webmail, CRM entre outros, através desta camada.

2.1.5 Modelos de Implantação

De acordo com (MELL, 2011) um ambiente de computação em nuvem pode ser implementado de 4 formas:

- a) Nuvem privada: é provisionada exclusivamente para o uso de uma organização;
- b) Nuvem pública: normalmente disponibiliza a infraestrutura da nuvem pela internet. Não possui um público alvo;
- c) Nuvem comunitária: provisiona os recursos de forma compartilhada para organizações com objetivos em comum; e
- d) Nuvem híbrida – composta por um ou mais tipos de implementação (público, privado e comunitária).

Nesse trabalho será abordado o modelo de serviço IaaS, que pode ser encontrado em plataformas tanto de código aberto ou de código proprietário. Recentemente a plataforma Openstack, de código aberto, vem se destacando e ganhando a adoção de grandes empresas, por exemplo IBM, HP, Dell e RedHat.

2.2 VIRTUALIZAÇÃO

A virtualização é uma tecnologia que foi desenvolvida pela empresa IBM na década de 1960 e usada inicialmente em *Main Frames* na tentativa de compartilhar grandes e caros equipamentos em partes menores, porém, na época, a tecnologia se restringiu a estes grandes equipamentos. Estudos e testes realizados naquela década proporcionaram tal tecnologia de forma que, atualmente, ela tem tomado proporções maiores e se mostrado como uma das melhores soluções à tecnologia da informação, quando se refere a redução de custos.

O aumento de capacidade de processamento de computadores de pequeno e médio porte, associado à redução do custo dos equipamentos atuais e o surgimento

de softwares que viabilizam a utilização dessa tecnologia, vem fazendo com que a virtualização seja empregada com uma força sem precedentes e se apresenta com destaque na hora de arquitetar e planejar futuras instalações e migrações de equipamentos e tecnologias.

Segundo Burger (2012), a tecnologia de virtualização é possivelmente umas das singularidades mais importantes atualmente na TI e está fazendo com que a indústria de computadores faça uma revisão nos conceitos operacionais que englobam hardware e software. O crescimento do conhecimento sobre as vantagens fornecidas com a tecnologia de virtualização foi impulsionado por fatores econômicos, escassez de recursos, regulamentações governamentais e maior competitividade entre as empresas.

O autor ainda afirma que a virtualização está sendo usada por um crescente número de organizações visando redução de custos que envolvem consumo de energia dos equipamentos, consumo dos equipamentos de ar condicionado, espaço útil nos edifícios para alocação de servidores, entre outras. Segundo o próprio autor, a virtualização fornece alta disponibilidade para aplicações críticas, dinamiza projetos de implementações, migrações, ajuda a simplificar as operações de TI e permite que organizações respondam mais rapidamente às mudanças de necessidades do mercado.

Segundo a empresa Dell (2011), a virtualização permite que você execute múltiplas aplicações e sistemas operacionais independentes em um único servidor. Adicionalmente, administradores podem rapidamente mover cargas de trabalho de um espaço virtual para outro, facilmente priorizando as necessidades dos negócios enquanto maximiza os recursos do servidor. Isto é possível através de softwares de virtualização como o ESXi, da VMware, Hyper-V, da Microsoft e do XenServer, da CITRIX, que são capazes de monitorar o uso do hardware e possibilitam fazer o balanceamento das máquinas virtuais (VMs) com o recurso de migração ao vivo, movendo-as de um servidor físico para outro de acordo com a demanda de sua aplicação.

Segundo a IBM, virtualizar aperfeiçoa a distribuição de carga dos serviços e flexibiliza o gerenciamento dos recursos (IBM, 2011).

Cada vez mais as empresas estão buscando formas de reduzir os custos e complexidade com o ambiente de TI. A virtualização tem se tornado uma ferramenta chave para uma estratégia eficiente na busca destes objetivos.

2.2.1 Hypervisor (VMM)

O *Hypervisor*, ou Monitor de Máquina Virtual (VMM), ou seja é uma camada de software entre o hardware físico e o sistema operacional convidado interagindo entre eles. É o responsável por fornecer ao sistema operacional convidado a abstração da máquina virtual, ou seja fornecer recursos compartilhados do hardware físico para os sistemas operacionais convidados. O *Hypervisor* controla o acesso dos sistemas operacionais convidados aos dispositivos de hardware. É importante salientar que o VMM não executa em modo usuário, pois é ele que deve executar, ou simular a execução, das instruções privilegiadas requisitadas pelo sistema operacional convidado (UFRJ, 2008).

O *Hypervisor* selecionado para o desenvolvimento deste trabalho é o XenServer, desenvolvido pela Citrix e baseado no projeto de código aberto Xen.

2.2.2 XenServer

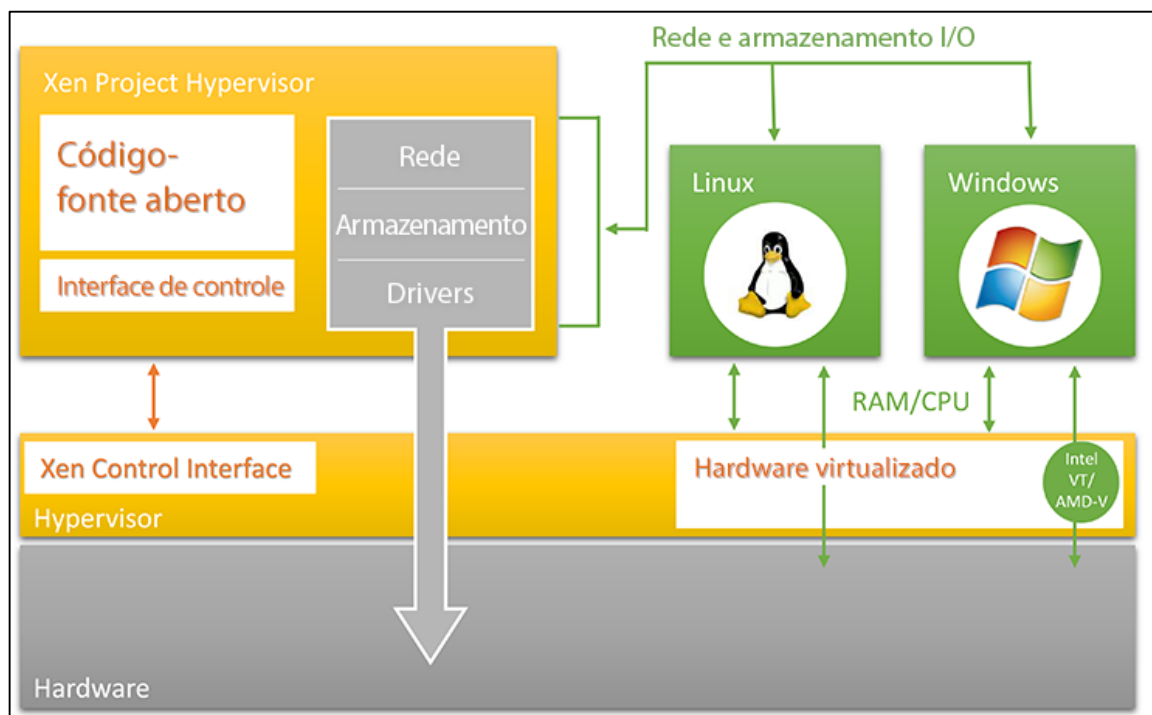
O XenServer é uma solução de infraestrutura virtual extremamente completa com um *Hypervisor* de 64-bits, possui um console intuitivo e de fácil gerenciamento do ambiente virtualizado, possui capacidades de *livemigration*, ou migração de máquinas virtuais em tempo real sem a necessidade de paradas e com as ferramentas necessárias para converter as grandes cargas de trabalho existentes de um ambiente físico para um ambiente virtualizado, sem maiores esforços, permite também as empresas criarem e gerenciarem uma quantidade ilimitada de servidores e máquinas virtuais, que podem rodar com segurança a partir de um console único de gerenciamento, facilitando e centralizando a gestão e o suporte. Com base no design de código aberto Xen, altamente confiável e seguro, fornece uma performance de aplicativos quase nativa, além de permitir uma fácil configuração de servidor, Storage e Network.

“O XenServer é uma solução de classe empresarial comprovada para computação em nuvem e virtualização de servidores, que reduz drasticamente os custos do datacenter ao transformar os ambientes de TI estáticos e complexos em data centers virtuais, dinâmicos e fáceis de gerenciar. Oferece capacidades de gerenciamento avançadas para integrar e automatizar os datacenters virtuais por uma fração do custo de outras soluções.” (CITRIX, 2011, p. 2).

Tomando como base a definição da CITRIX, pode-se observar que o foco é o meio corporativo e construção de ambientes de nuvem, transformando data centers estáticos em ambientes virtuais e dinâmicos, tendo assim um maior aproveitamento do hardware, centralizando o gerenciamento e reduzindo drasticamente os custos de um data center.

Na figura 5 podemos observar como é a arquitetura do VMM XenServer:

Figura 5 - Diagrama da arquitetura de virtualização



Fonte:

http://www.citrix.com.br/_onelink_/citrix/en2ptbr/images/content/citrix/en_us/products/xenserver/tech-info/_jcr_content/par/highlightedcontent/highlightedcontentpar/imageframe/image2.img.png/1406564893375.png

Podemos observar na imagem acima que a comunicação do hardware com o sistema operacional convidado passa sempre pelo *Hypervisor XenServer*, passando pelo hardware virtual requisitando acesso ao sistema operacional convidado e então interagindo como um dispositivo de entrada e saída no sistema operacional convidado. Esta relação é simples observando na imagem, contudo há uma série de processos disseminados e interdependentes que devem ser executados quase que instantaneamente reduzindo a perda de desempenho no ambiente virtual.

2.2.3 Citrix XenCenter

O Citrix XenCenter é um software de interface gráfica para usuários que utilizam plataforma windows nativo para o gerenciamento do Citrix XenServer. Este software permite a instalação completa de máquinas virtuais, configuração e gerenciamento do ciclo de vida, acesso ao console das VMs, gerenciamento de rede, incluindo VLANs e redes internas, ligado e interfaces de rede físicas (NICs) dedicadas.

2.3 OPENSTACK

Openstack pode ser entendido como um conjunto de softwares de código aberto que teve seu desenvolvimento iniciado principalmente pela Rackspace (Provedor de infraestrutura de TI americano) e NASA (Agência Espacial Americana). A contribuição da NASA com este projeto foi fornecer a plataforma de gerenciamento de infraestrutura “*Nebula*” que veio a se tornar base para o projeto “*Nova*” para gestão computacional, já a Rackspace contribuiu fornecendo sua plataforma de armazenamento “*Cloud Files*” que veio a se tornar o “*Object storage*” do *Openstack*. A partir deste momento diversas empresas vieram a se unir ao projeto formando um consórcio que envolve os grandes da tecnologia no mundo como Dell, IBM, Citrix e Canonical.

Openstack Cloud Software é um Sistema Operacional na Nuvem capaz de gerenciar componentes de múltiplas infraestruturas virtualizadas. É chamado assim porque realiza o mesmo papel de um sistema operacional, porém em grande escala na web. O OpenStack controla grandes pools de computação (recursos agrupados para atender diversos usuários), armazenamento e recursos de rede por meio de um Data Center, tudo gerenciado através de um painel de interface web com controle de administração para configurar recursos entre administradores e usuários. (SOFTWARE LIVRE, 2013)

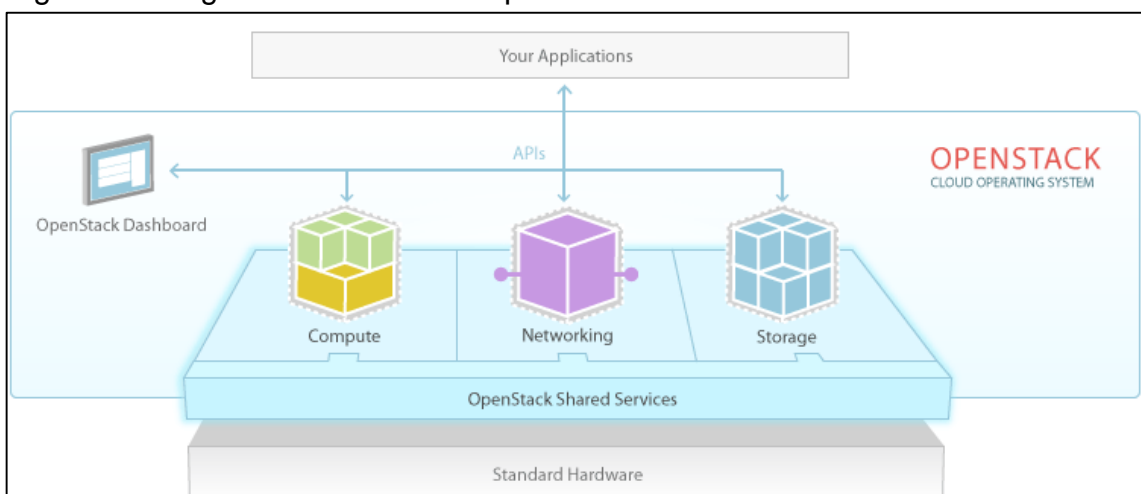
Tomando como base a definição do Software Livre, pode-se afirmar que o *Openstack* é um gigante sistema operacional, pois tem a capacidade de gerir um grande volume de servidores virtualizados e agrupados a fim de atender a grandes demandas de software, plataformas e hardwares existentes na atualidade, visto que a facilidade para ofertar recursos e o gerenciamento simplificado através de uma interface web que lhe permite acesso de qualquer navegador o torna automaticamente uma das melhores opções para o gerenciamento de ambientes virtualizados e heterogêneos, ou seja diferentes *hypervisors*.

Segundo (IBM, 2014) o *Openstack* pode ser visto como um grande sistema operacional na nuvem, onde, por meio de várias APIs (*Application Programming Interface*) desenvolvidas, é possível orquestrar todos os recursos virtualizados por ele.

Esse projeto foi desenvolvido inicialmente por uma força tarefa no ano de 2010 com o objetivo de criar um padrão aberto para habilitar serviços de nuvem com base em elementos de hardware padrão. A partir dessa data vários fornecedores de tecnologia também começaram a desenvolver produtos aderentes e suportado pela plataforma *Openstack*. Atualmente, a plataforma conta com a colaboração de mais de 200 empresas de hardware e software.

O *Openstack* disponibiliza um ambiente de armazenamento e processamento virtual sob o modelo *self service*, ou seja, a infraestrutura cresce sob demanda e por consequência, permite a cobrança somente do que foi provisionado. Essa característica maximiza a utilização dos recursos.

Figura 6 - Diagrama do Software OpenStack



Fonte: <http://www.openstack.org/themes/openstack/images/openstack-software-diagram.png>

Conforme o Diagrama do Software *Openstack* (figura 6) é composta de muitas partes móveis diferentes, o mesmo possui 3 projetos principais que são os pilares para a estrutura desta excelente ferramenta, os projetos são o *Compute*, *Networking* e *Storage*, tendo como suporte uma série de recursos e serviços compartilhados permitindo serem acessados através do *Dashboard* ou APIs, disponibilizados através do *Networking* e processados através do projeto *Compute*. Mas a comunidade *Openstack* identificou nove principais componentes que fazem parte do núcleo do *Openstack*, que são distribuídos como parte de qualquer sistema *Openstack* e oficialmente mantido pela comunidade *Openstack*.

a) **Nova** é parte do projeto *Compute* e é o motor de computação principal por trás *Openstack*. É um "controlador de recursos computacionais", que é usado para implantar e gerenciar um grande número de máquinas virtuais e outras instâncias para lidar com tarefas de computação.

b) **Swift** é um sistema de armazenamento de objetos e arquivos. Ao contrário do que a ideia tradicional que se refere a arquivos pela sua localização em uma unidade de disco, os desenvolvedores podem, em criar referências a um identificador único apontando ao arquivo ou informação e deixar *Openstack* decidir onde armazenar esta informação. Isso faz a escalabilidade parecer fácil, pois os desenvolvedores não têm a preocupação com a capacidade em um único sistema por trás do software. Ele também permite que o sistema, em vez do desenvolvedor se

preocupar com a melhor forma de se certificar de que os dados são copiados em caso de falha de uma conexão de máquina ou rede.

c) **Cinder** é um componente de armazenamento em bloco, o que é mais análogo à noção tradicional de um computador ser capaz de acessar locais específicos em uma unidade de disco. Esta é uma das formas mais tradicionais de acesso a arquivos e pode ser importante em cenários em que a velocidade de acesso aos dados é a considerada mais importante.

d) **Neutron** é parte do projeto *Networking*, fornece a capacidade de rede para *Openstack*. Isso ajuda a garantir que cada um dos componentes de uma implantação *Openstack* possam se comunicar umas com as outras de forma rápida e eficiente.

e) **Horizon (Dashboard)** é o painel atrás *Openstack*, esta é a única interface gráfica para *Openstack*, portanto, para os usuários que querem conhecer o programa, este pode ser o primeiro componente que eles realmente irão “ver”. Os desenvolvedores podem acessar todos os componentes do *Openstack* individualmente através de uma interface de programação de aplicativo (API), mas o painel fornece aos administradores de sistema uma visão geral do que está acontecendo na nuvem, e para gerenciá-los conforme necessário.

f) **Keystone** é um dos serviços compartilhados de suporte ao *Openstack* e fornece serviços de identidade. É essencialmente um ponto central de todos os usuários da nuvem *Openstack*, mapeados com seus respectivos serviços contratados na nuvem e que possuem permissões para o uso. Ele oferece vários meios de acesso, ou seja, os desenvolvedores podem facilmente mapear os seus métodos de acesso de usuário existentes através do *Keystone*.

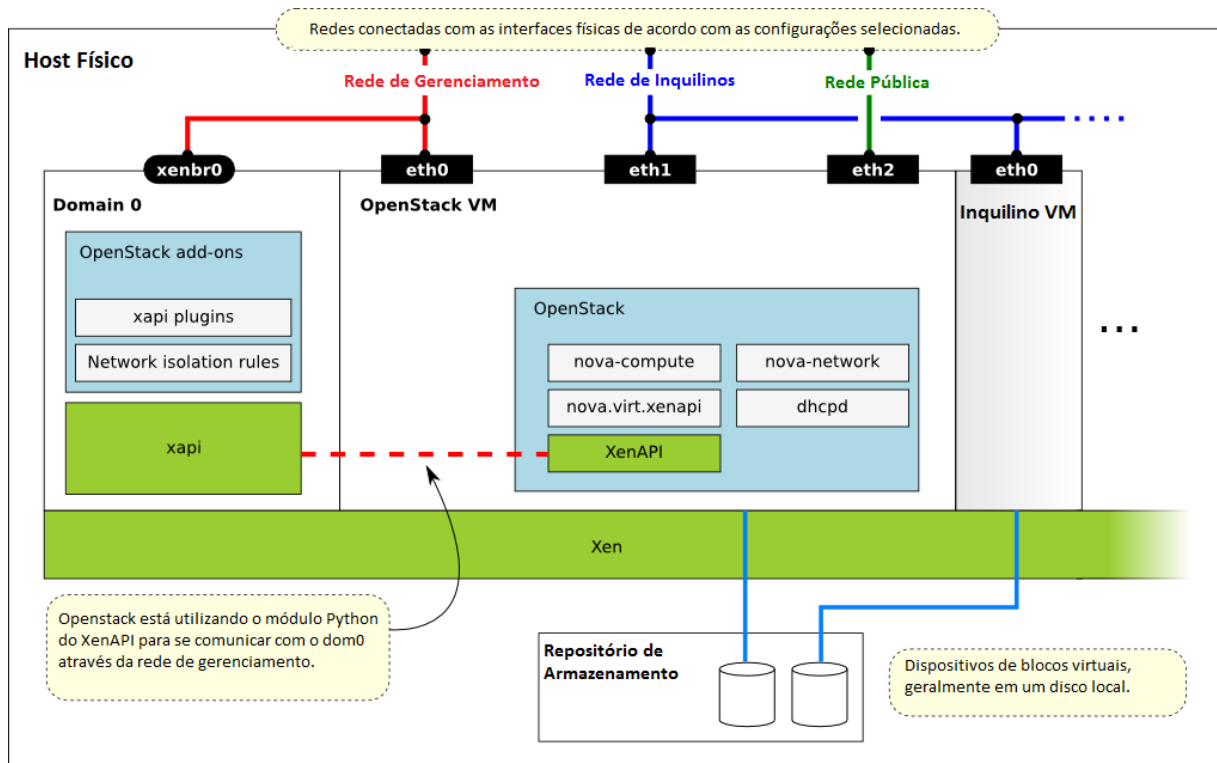
g) **Glance** fornece serviços de imagem para *Openstack*. Neste caso, as imagens virtuais (ou cópias) de discos rígidos. O *Glance* permite que essas imagens sejam utilizadas como modelos quando necessário for implantar novas instâncias de máquinas virtuais.

h) **Ceilometer** fornece serviços de telemetria, que permitem a nuvem fornecer um serviço de faturamento para usuários individuais da nuvem. Ele também mantém uma verificação e medição de uso do sistema de cada usuário de cada um dos diversos componentes de uma nuvem *Openstack*.

i) **Heat** é o componente de orquestração do *Openstack* e permite aos desenvolvedores armazenar os requisitos de um aplicativo em nuvem em um arquivo

de configuração que define os recursos necessários para a execução de uma determinada aplicação. Desta forma, ele ajuda a gerenciar a infraestrutura necessária para a execução de um serviço na nuvem (OPENSOURCE, [2008]):

Figura 7 - Arquitetura de Implantação do sistema *Openstack*



Fonte: http://docs.openstack.org/grizzly/openstack-compute/install/apt/content/figures/5/figures/xenserver_architecture.png

2.3.1 Arquitetura do Openstack

A arquitetura do *Openstack* é modular e contempla os seguintes projetos: Nova, Glance, Swift e Cinder. Os dois primeiros tratam das questões de processamento, computação e serviço de imagens e os dois últimos endereçam questões de armazenamento. As duas versões de 2013 (Grizzly e Havana) trouxeram grandes melhorias e funcionalidades, principalmente nas áreas de rede, armazenamento, *autoscaling* e monitoramento. A primeira de 2014 (*Icehouse*), trouxe novas funcionalidades que amadurecem ainda mais essa plataforma. Em sua nona versão, os destaques ficam para o grande aumento da estabilidade, a integração melhorada entre seus módulos e o projeto *Trove*, que provê *database as a service*. Está prevista

ainda uma nova versão para 2014, com o nome Juno. Suas principais novidades são a de *bare metal as a service* (projeto *Ironic*), um novo serviço de mensagens ou *messagequeue* (projeto Marconi) e também um de *data processing* que poderá ser utilizado para provisionamento de *clusters Hadoop* (projeto Sahara).

O futuro do *Openstack* é promissor, com *roadmap* já muito amadurecido. Em tempos de forte pressão sobre a eficiência e o retorno de investimento em infraestrutura de TI, a plataforma *Openstack* pode ser considerada como uma opção. Invista tempo no estudo desse assunto, pois análogo ao *Openstack*, onde a flexibilidade é o caminho mais certo a seguir, a adaptação às mudanças é fundamental.

2.4 NAS

NAS (*Network Attached Storage*) ou Armazenamento conectado à rede, podemos entender como um dispositivo servidor de armazenamento de dados em rede. O dispositivo NAS possui acesso lógico através de cabo de rede assumindo assim entrada para a internet, onde os arquivos podem ser acessados através de um endereço de rede configurado no mesmo. Normalmente quando se atribui este tipo de função para o equipamento o mesmo não recebe outra atribuição, pois mesmo tendo recursos como processador, placa mãe e memória que possibilitam a instalação de aplicativos, suas funções são restritas ao armazenamento de dados.

Utilizando os protocolos mais populares em sistemas UNIX, ou CIFS/SMB (*Common Internet File System/Server MessageBlock*) para ambientes Windows, assim como o bem conhecido FTP. O NAS de uso doméstico é baseado em processadores baratos rodando uma versão embarcada de um sistema Linux. Existem diversas alternativas de código aberto que permitem instalações em ambientes reduzidos, como o *FreeNAS* que é o escolhido utilizar neste trabalho. (ADRENALINE, 2012)

2.4.1 Freenas

O *FreeNAS* é um sistema operacional de armazenamento de software baseado em FreeBSD, que possui suporte a todos os principais protocolos de armazenamento do mercado. Este é uma das grandes diferenças e a razão de ter escolhido esta

ferramenta: o *FreeNAS* é focado em protocolos de rede de armazenamento. Isto inclui, mas não está limitado a *Common Internet File System* (CIFS) que é utilizado para a rede Windows, FTP, NFS e iSCSI.

Além de flexível, pois pode ser instalado diretamente no hardware ou dentro de uma máquina virtual (VM). Como o próprio nome já diz ele é livre, ou seja seu código é aberto. Na verdade, recursos como replicação, de duplicação, e outras funcionalidades inteligentes que impulsionam essas soluções agregam um valor incrível para os administradores que buscam uma solução gratuita.

O foco principal do FreeNAS é ajudar usuários de software livre através de um fácil e rápido gerenciamento de armazenamento. (TECH REPUBLIC, 2011)

2.5 LINUX

O Linux é um sistema operacional livre e é uma reimplementação das especificações POSIX (padronização da IEEE, Instituto de Engenharia Elétrica e Eletrônica) para sistemas com extensões *System V* e BSD. Isso significa que o Linux é bem parecido com Unix, mas não vem do mesmo lugar e foi escrito de outra forma.

O desenvolvedor Linus Torvalds, quando desenvolveu o Linux, não tinha a intenção de ganhar dinheiro e sim fazer um sistema para seu uso pessoal, que atendesse suas necessidades. O estilo de desenvolvimento que foi adotado foi o de ajuda coletiva. Ou seja, ele coordena os esforços coletivos de um grupo para a melhoria do sistema que criou. Milhares de pessoas contribuem gratuitamente com o desenvolvimento do Linux, simplesmente pelo prazer de fazer um sistema operacional melhor.

Atualmente o Linux está sob a licença GPL que permite que qualquer um possa usar os programas que estão sob ela, com o compromisso de não tornar os programas fechados e comercializados. Ou seja, você pode alterar qualquer parte do Linux, modificá-lo e até comercializá-lo, mas você não pode fechá-lo (não permitir que outros usuários o modifiquem) e vendê-lo (INFOWESTER, 2011).

[...] no caso de um sistema operacional, o código-fonte - as instruções de programação implícitas no sistema - é livre. Qualquer pessoa pode melhorá-lo, transformá-lo, explorá-lo. Porém, essas melhorias, transformações e explorações precisam ser disponibilizadas livremente. Pense Zen. O projeto não pertence a ninguém e pertence a todos. Ao abri-lo a todos, há um aperfeiçoamento rápido e contínuo. Com equipes de colaboradores trabalhando em paralelo, os resultados podem acontecer muito mais

depressa e com muito mais sucesso do que se estivessem sendo conduzidos a portas fechadas. (Torvalds; Diamond, 2001, p. 261).

Tomando como base a literatura de Torvalds e Diamond o Linux tornou-se uma solução para o mundo da tecnologia, devido a ideologia de compartilhar conhecimento. Isto permite que qualquer um com conhecimento equivalente em programação possa explorá-lo e torná-lo cada vez mais robusto, conforme as necessidades de ambiente, de arquitetura e de hardware. A potência e a exploração da computação em nuvem é base do resultado da velocidade com que grupos de desenvolvimento colaborativo fizeram para a evolução da tecnologia através de sua disseminação na internet, ou seja de portas abertas.

2.6 DEVSTACK

O Devstack é um script que cria rapidamente um ambiente de desenvolvimento *Openstack*. Ele também pode ser usado para demonstrar a execução dos serviços *Openstack* e fornecem exemplos de como usá-los a partir de uma linha de comando, estes exemplos de exercícios foram desenvolvidos tomando como base exemplos simples de uso e tornou-se muito útil para realização de testes rápidos de sanidade para a instalação do de sanidade rápida para a instalação do *Openstack*. Estes testes permanecem com a prioridade principal que é o desenvolvimento e evolução do script *Devstack*.

O DevStack não é e nunca foi destinado a ser um instalador principal do OpenStack, ele evoluiu para suportar um grande número de opções de configurações e plataformas alternativas, assim como serviços de apoio (DEVSTACK, 2014).

2.6.1 MISSÃO

A missão da Devstack é fornecer e manter as ferramentas utilizadas para a instalação dos serviços principais do Openstack cuja a fonte está armazenada no sistema de repositório *Github* com o projeto denominado “master” que atualmente é o sistema de repositório mais adequado para o desenvolvimento e testes operacionais. Ele também demonstra e documenta exemplos de configuração e execução de serviços, bem como o uso de comandos por clientes no console (DEVSTACK, 2014).

2.7 UBUNTU LINUX

O Ubuntu é um sistema operacional desenvolvido pela *Canonical* em conjunto com a comunidade Linux, e por profissionais da área. É considerado um dos sistemas operacionais mais perfeitos e completos para o uso em notebooks, desktops e servidores, pois possui todas as ferramentas necessárias para um sistema operacional que pode ir desde um editor de textos até um servidor web.

O discurso principal da empresa Canonical é de que o *Ubuntu* é e sempre será **gratuito**, ou seja sem encargos com licenças assim com todos as distribuições que usufruem do licenciamento GNU Linux.

A cada seis meses é lançada uma nova versão, o que significa que vocês sempre terão disponíveis as mais atualizadas versões dos principais softwares utilizados no mesmo. O *Ubuntu* é desenvolvido visando segurança, as atualizações de segurança são disponibilizadas por pelo menos 18 meses para todas as versões. Com a versão de Longo Tempo de Suporte (LTS) pode-se obter três anos de suporte para desktops, e cinco anos para servidores. Na versão servidor é possível ter tudo o que precisa para ter o servidor funcionar sem softwares desnecessárias.

3 METODOLOGIA

3.1 DELINEAMENTO DA PESQUISA

Este capítulo está estruturado de forma que apresenta inicialmente o delineamento da pesquisa, ou seja, como ela se classifica cientificamente. Logo após foi apresentado o modelo de computação em nuvem que este trabalho propõe. Assim como a descrição de itens que este trabalho irá responder e outros que não serão abordados.

Esta pesquisa possui uma abordagem quali-quantitativa. Há aspectos quantitativos na mensuração de custos, espaço de armazenamento, disponibilidade e agilidade nos processos de TI, entre outros dados. A análise principal baseia-se em aspectos qualitativos que são os aspectos que demonstram a facilidade, a acessibilidade e a portabilidade do uso da computação em nuvem.

O método de pesquisa utilizado foi um estudo de caso em implantação de uma infraestrutura de TI em nuvem para um provedor deste serviço.

Esta pesquisa tem por objetivo fornecer um modelo de tecnologia, denominada computação em nuvem que tem como principal propósito, disponibilizar poder de processamento, armazenamento de dados, conectividade, plataformas, aplicações e serviços de forma segura, escalável e de alta disponibilidade.

Não será descrito nesta pesquisa como montar o hardware e/ou instalar o Citrix XenServer, mas sim apresentar o processo de instalação da ferramenta *Openstack* de forma a automatizar o processo de implantação, separando a inteligência da arquitetura do processo de instalação, assim como os principais recursos e arquitetura necessária para implantação da tecnologia de computação em nuvem para que, no final, a instalação torne-se personalizada e seja distribuída uma máquina virtual em um Sistema Operacional de base Linux de fácil instalação.

A implantação do *Openstack* deve ter um mínimo de 3 *nodes* (nós), o *controller* (controlador), a *network* (rede) e o *compute* (computação). Visto este ambiente é importante observar que deste modo não teria capacidade para suprir uma implantação de alta disponibilidade como uma plataforma de computação em nuvem, onde seria necessário ter a capacidade de *failover* (superar falhas) de funções no modo de espera dos nós, assim quando houver uma falha ou uma queda de energia ou uma manutenção necessária. Contudo, para este trabalho, vou apresentar um

modo que é chamado de *all-in-one* (tudo em um), neste modo há todos os nós: controlador, rede e computação serão instalados e serão executados no mesmo nó. Embora o modo de instalação seja utilizado em apenas um nó, será possível representar de forma completa o uso de uma plataforma de computação em nuvem de infraestrutura como um serviço.

3.2 HARDWARE

Para realização deste ambiente foram utilizados os seguintes recursos:

Hardware do Servidor 1:

Processador: Genuine Intel Core i5-3330 CPU 3.00GHz,

Speed: 2993 MHz.

Memória: 8192 MB (megabyte)

Armazenamento: um disco Seagate de 1 TB (terabyte).

Hypervisor: Citrix XenServer versão 6.2 compilação 70446c

Hostname: xscloud01.horizon.lan.

Endereço IP: 192.168.10.250/24

Armazenamento Compartilhado:

Processador: Genuine Intel Core i5-3330 CPU 3.00GHz,

Speed: 2993 MHz.

Memória RAM: 16384 MB (megabyte)

Armazenamento: um disco Seagate de 1 TB (terabyte).

Sistema Operacional: FreeNAS

Hostname: stcloud01.horizon.lan.

Endereço IP: 192.168.10.190/24

Rede:

Equipamento: Mikrotik Router Board 750GL

Endereço IP: 192.168.10.1/24

Portas: 5

3.2 INSTALAÇÃO DE PACOTES XENAPI

Após a instalação do Citrix XenServer foram necessários alguns procedimentos no servidor *Hypervisor* Citrix XenServer para que o mesmo recebesse instruções do Openstack, acionando assim a API do XenServer XenAPI, que estão disponíveis no repositório de código Github através do link:

<https://github.com/openstack/nova/tree/master/>.

Abaixo os procedimentos realizados para instalação dos plug-ins XenAPI:

```
[root@xs01]# mkdir /openstack
[root@xs01]# cp -avf * /nova-master.zip /openstack
[root@xs01]# cd /openstack ;unzip nova-master.zip
[root@xs01 openstack]# cd nova-master/
[root@xs01 plugins]# cp -avf * /etc/xapi.d/plugins/
```

O diretório “/etc/xapi.d/plugins/” contém arquivos que são necessários para o apoio XenAPI. Estes devem possuir permissões de execução para funcionar corretamente, para isso o comando abaixo foi realizado no servidor XenServer:

```
[root@xs01 plugins]# chmod a+x /etc/xapi.d/plugins/*
```

O Isolamento de VIF ou (*Virtual Interface*) de rede é fundamental na utilização do Openstack, pois isso permite que sejam separadas as redes de uma nuvem pública de uma nuvem privada, assim como a restrição no acesso de determinados recursos, através da rede, permitindo assim mais segurança ao ambiente. Contudo, para que este isolamento seja realizado, é necessário adicionar um plugin de script contendo estas restrições conforme comando abaixo:

```
[root@xs01 nova-master]# cp -avf /openstack/master-nova/plugins/xenserver/networking/etc/xensource/scripts/vif_rules.py /etc/xensource/scripts/
```

Para que toda comunicação entre o XenServer e o Openstack ocorra é necessário estabelecer uma comunicação segura através do protocolo SSH onde será necessário realizar uma configuração criando chaves públicas no tamanho de 2048 bits e adicionando ao servidor *Openstack*. Para permitir que recursos como

redimensionamento de um disco VHD seja copiado usando ssh e rsync, foi necessário configurar o acesso sem senha entre todos os XenServer que fazem parte de seu ambiente *cloud*, conforme instruções abaixo:

```
[root@xs01 nova-master]# cd ~
[root@xs01 ~]# cd .ssh
[root@xs01 .ssh]# ssh-keygen -t rsa -b 2048 -N ""
[root@xs01 .ssh]# ls
[root@xs01 .ssh]# chmod 700 ~/.ssh
[root@xs01 .ssh]# chmod 600 ~/.ssh/id_rsa
```

A funcionalidade 'redimensionamento' depende de um diretório específico no XenServer, por isso, foi necessário realizar a criação do diretório "/imagens", conforme instruções abaixo:

```
[root@xs01 ~]# mkdir /imagens
```

Para o uso de todos os recursos disponíveis na comunicação entre o XenServer e o *Openstack* foi necessário realizar a instalação dos pacotes de instalação da linguagem *Python 2.6*.

A melhor maneira é usar o mais recente pacote de EPEL, disponibilizado pela comunidade *CentOS* e *Fedora*.

```
[root@xs01 ~]# wget http://dl.fedoraproject.org/pub/epel/5/i386/epel-release-5-4.noarch.rpm
[root@xs01 ~]# rpm -Uvh epel-release-5-4.noarch.rpm
[root@xs01 ~]# yum install python26-distribute
```

Este procedimento irá instalar os seguintes pacotes: *libffi*, *Python26*, *python26-distribute*, *Python26-libs*

Neste momento o yum pode ficar muito confuso entre a versão Citrix e a versão EPEL. Para evitar desastres ao executar "yum upgrade", é melhor desabilitar o repositório EPEL e excluir o rpm que você baixou:

```
[root@xs01 ~]# rpm -ev epel-release
[root@xs01 ~]# rm -f epel-release-5-4.noarch.rpm
```

Instalação do *python-swiftclient*

Agora que já temos o pacote do *python* 2.6, podemos realizar a instalação de alguns serviços públicos do *python*.

```
[root@xs01 ~]# easy_install-2.6 pip
[root@xs01 ~]# pip install python-swiftclient python-keystoneclient
```

3.3 INSTALAÇÃO DO DEVSTACK

Começando com o XenServer e Devstack

Conforme apresentado no referencial teórico, será utilizado neste trabalho o modo de implantação do Openstack denominado Devstack onde é a soma de uma instalação do *Ubuntu* LTS 14.04.1 com ambiente de desenvolvimento *Openstack*.

XenServer é o *hypervisor* escolhido para desenvolver este trabalho, onde o mesmo está instalado diretamente no hardware. O serviço do *Openstack* está configurado para ser executado em uma máquina virtual chamado “OS domU” no host XenServer. A máquina virtual utiliza o *toolstack* XAPI para comunicar-se com o host através de uma conexão de rede denominada (MGT_BRIDGE_OR_NET_NAME). Junto a este escopo há o arquivo “localrc” fornecido pelo próprio *Openstack* para facilitar a construção de um ambiente básico de computação em nuvem com o *Openstack*.

3.3.1 Requisitos Esquisitos

- a) Uma rede conectada com a internet e um servidor DHCP na mesma.
- b) Um host servidor XenServer conectado na mesma rede. Esta rede será usada como a rede de gerenciamento do *Openstack*. A Rede VM e a rede pública não serão conectados a todas interfaces físicas, apenas novas redes virtuais serão criadas através do script de instalação do *openstack* “install_os_domU.sh”.

3.3.2 Processo Macro de Instalação

1. Instalar XenServer;
2. Baixe Devstack para XenServer;

3. Personalize localrc; e
4. Comece install_os_domU.sh roteiro.

3.3.3 Roteiro de Instalação (install_OS_domU.sh)

1. Instalação dos *plugins* XenAPI no servidor XenServer;
2. Criar as redes e nomeá-las, caso não exista;
3. Realizar o processo de *Net-install*, ou seja realizar o *download* de uma instalação mínima de uma máquina virtual Ubuntu LTS 14.04.1 (Esta será nossa *appliance* ao termino deste trabalho), com uma interface de rede;
4. eth0 - Conetado a UBUNTU_INST_BRIDGE_OR_NET_NAME, o padrão é MGT_BRIDGE_OR_NET_NAME;
5. Após o processo de instalação do Ubuntu estar concluída, a configuração de rede será modificada para;
6. eth0 - Interface de gerenciamento, conectado a interface MGT_BRIDGE_OR_NET_NAME. O Xapi será acessado através desta rede;
7. eth1 - Interface VM, conectado a interface VM_BRIDGE_OR_NET_NAME;
8. eth2 - Interface pública, conectado a interface PUB_BRIDGE_OR_NET_NAME; e
9. Inicialize o *devstack* criado dentro da máquina virtual Ubuntu com o *Openstack*.

3.3.4 Etapa 1: Instalar XenServer

Instalar XenServer, é importante que seja uma nova instalação, sem qualquer alteração na estrutura do sistema. O XenServer pode ser baixado gratuitamente a partir de: <http://www.xenserver.org/>.

A configuração XenServer IP depende de rede local, neste caso será utilizado o endereço IP 192.168.10.250/24 e como *gateway* padrão será utilizado o endereço 192.168.10.1, este endereço deve ser especificado no arquivo localrc. É importante se certificar se um *gateway* e um servidor de DNS está configurado para que seja executado o script install_os_domU.sh, pois o mesmo realiza uma conexão ao destino "github.com" para obter as imagens de instalação e o código-fonte.

3.3.5 Etapa 2: Baixar o devstack

Acessado o XenServer execute os seguintes comandos como *root*:

```
[root@xs01 ~]# wget --no-check-certificate https://github.com/openstack-dev/devstack/zipball/master
[root@xs01 ~]# unzip -o master -d ./devstack
[root@xs01 ~]# cd devstack*/
```

Esta etapa irá baixar os arquivos necessários do repositório do *Devstack* no *github*, descompactar o arquivo no formato zip e acessar o diretório descompactado anteriormente.

3.3.6 Etapa 3: Configurar o arquivo “localrc” localizado dentro do diretório “devstack”

Devstack usa o arquivo “*localrc*” para a configuração específica do usuário. Observe que o *XENAPI_PASSWORD* deve ser a sua senha de *root* no dom, ou seja no XenServer. Para criar o arquivo “*localrc*” com as configurações necessárias, execute o bloco de comandos abaixo com suas respectivas alterações para o preenchimento correto das configurações de instalação do host “*devstack*”, lembrando sempre do escopo definido.

```
[root@xs01 ~]# cat > ./localrc <<EOF
# At the moment, we depend on github's snapshot function.
GIT_BASE="http://github.com"
```

```
# Passwords
# NOTE: these need to be specified, otherwise devstack will try
# to prompt for these passwords, blocking the install process.
```

```
MYSQL_PASSWORD=citrix
SERVICE_TOKEN= citrix
ADMIN_PASSWORD= citrix
SERVICE_PASSWORD= citrix
RABBIT_PASSWORD= citrix
SWIFT_HASH="66a3d6b56c1f479c8b4e70ab5c2000f5"
# This will be the password for the OpenStack VM (both stack and root users)
GUEST_PASSWORD= citrix
```

```
# XenAPI parameters
# NOTE: The following must be set to your XenServer root password!
```

```
XENAPI_PASSWORD=maestrocloud3212
XENAPI_CONNECTION_URL="http://192.168.10.250"
VNC_SERVER_PROXYCLIENT_ADDRESS=192.168.10.250
```

```
# Download a vhd and a uec image
IMAGE_URLS="\
https://github.com/downloads/citrix-openstack/warehouse/cirros-0.3.0-x86_64-
disk.vhd.tgz,\
http://download.cirros-cloud.net/0.3.2/cirros-0.3.2-x86_64-uec.tar.gz"
```

```
# Explicitly set virt driver
VIRT_DRIVER=xenserver
```

```
# Explicitly enable multi-host for nova-network HA
MULTI_HOST=1
```

```
# Give extra time for boot
ACTIVE_TIMEOUT=45
EOF
```

3.3.7 Etapa 4: Executar o script `./install_os_domU.sh` a partir do diretório “tools/xen”

```
[root@xs01 ~]# cd tools/xen
[root@xs01 ~]# ./install_os_domU.sh
```

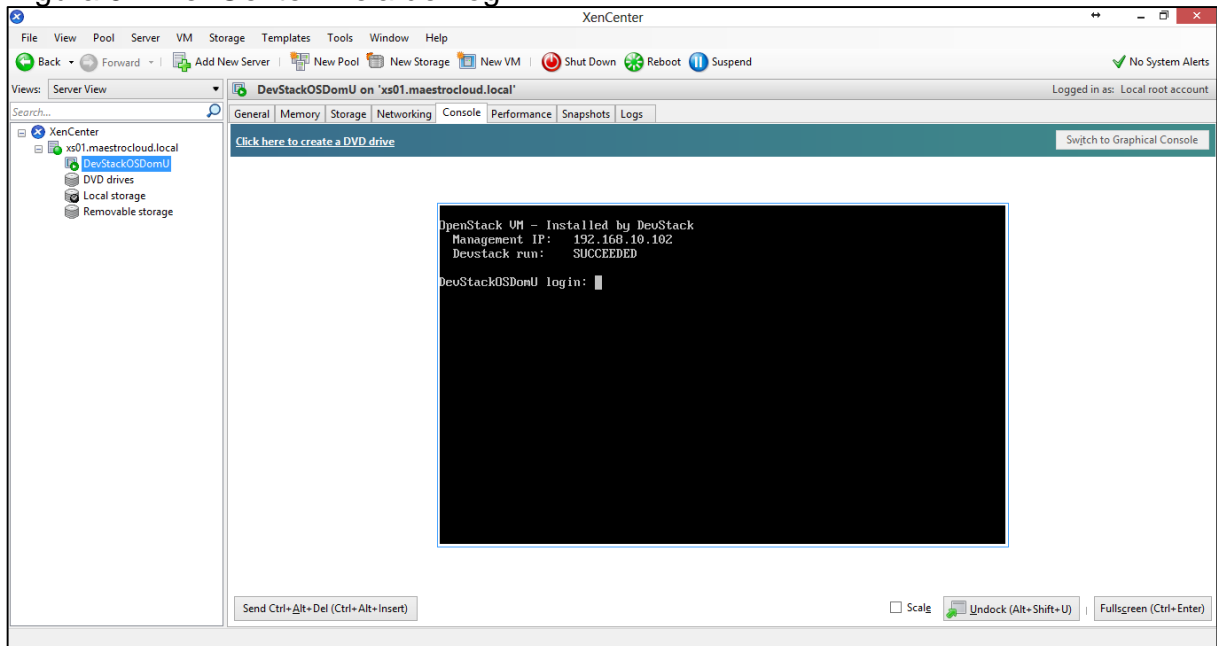
Figura 8 – Primeiro Login *Openstack*



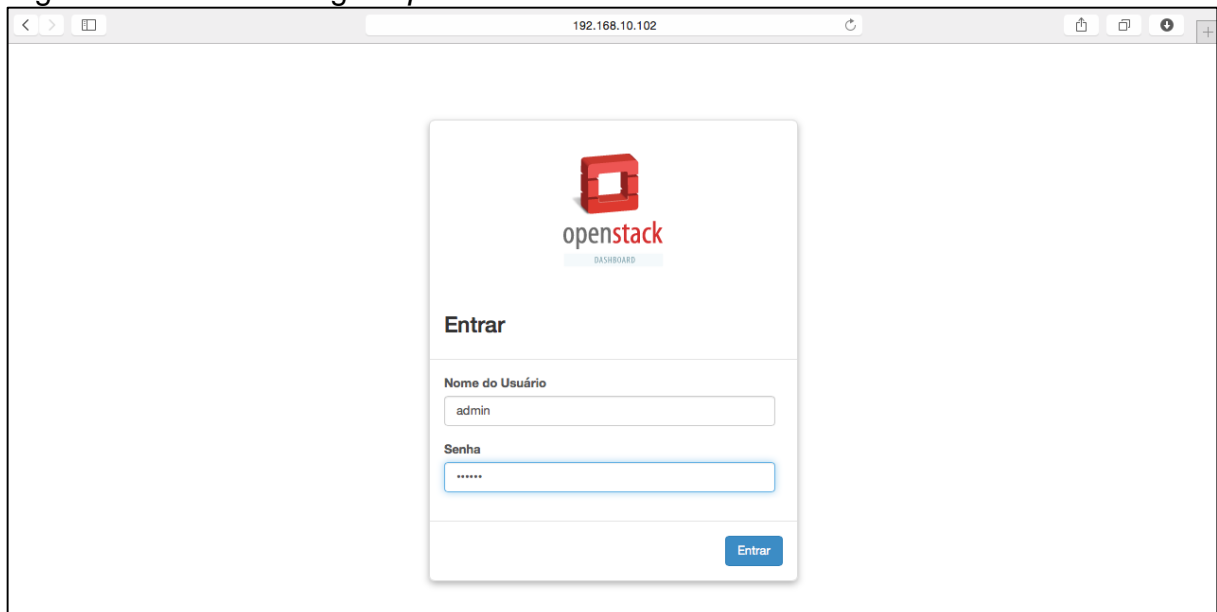
```
OpenStack VM - Installed by DevStack
Management IP: 192.168.10.102
Devstack run: SUCCEEDED
DevStackOSDomU login: █
```

Fonte: Desenvolvido pelo autor.

Figura 9 – XenCenter: Tela de Login



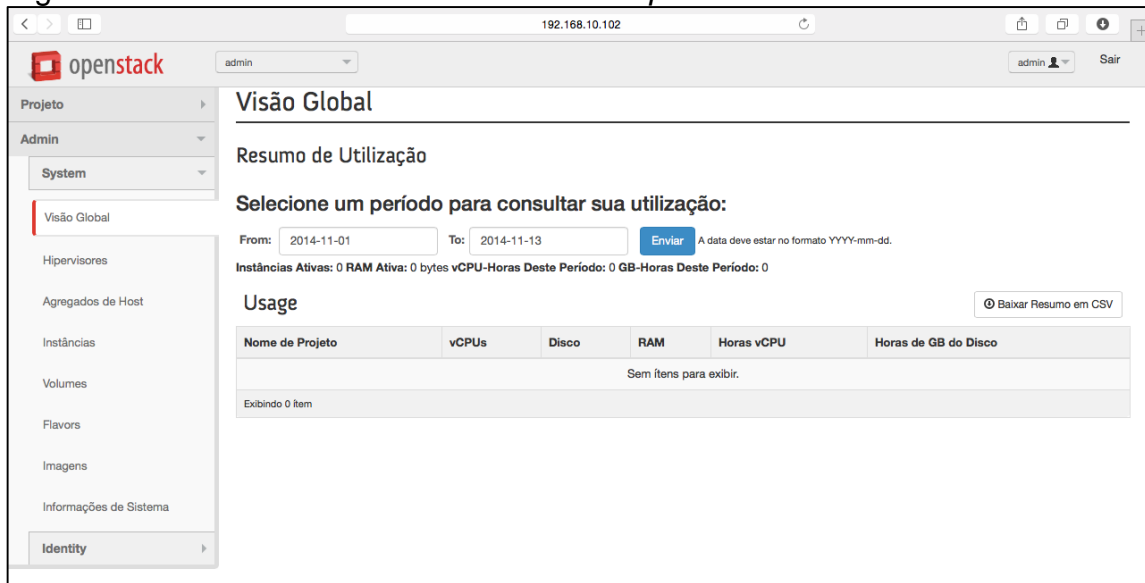
Fonte: Desenvolvido pelo autor.

Figura 10 – Tela de Login *Openstack*

Fonte: Desenvolvido pelo autor.

Agora acesse o sistema com o usuário admin e a senha “citrix” configurada em nosso arquivo de instalação do *Openstack* “localrc”.

Figura 11 – Tela de *Dashboard Horizon* do *Openstack*

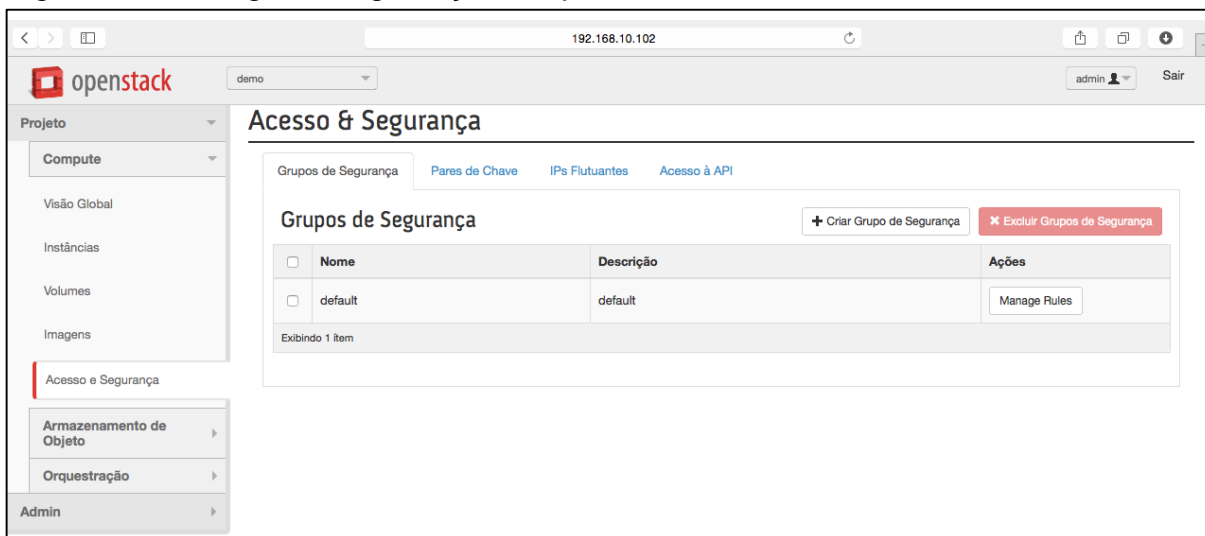


Fonte: Desenvolvido pelo autor.

A Figura 11 mostra como o *Openstack* já traz em sua tela principal o resumo de todos os recursos utilizados pelo sistema. Até este momento ainda não foi iniciada nenhuma instância no *Openstack*, pois é o primeiro acesso e por esta razão nenhum recurso está sendo utilizado.

a) Configurar Segurança do *Openstack*

Não menos importante do que qualquer outra etapa do projeto, é necessário analisar as regras e políticas de segurança para o ambiente e/ou escopo que se deseja, neste caso ficarão as regras que são adicionadas automaticamente pelo *Openstack*, mas lembrando que se adicionar um ambiente de produção não hesitar em gastar 1 hora do projeto analisando as regras de firewall. O grupo de regras "default" pode ser visualizado na imagem abaixo, incluindo IPv4 e IPv6, pois ambas são suportadas na plataforma *Openstack*.

Figura 12 - Configurar Segurança do *Openstack*

Fonte: Desenvolvido pelo autor.

b) Reiniciar *Openstack*

Como todo bom sistema operacional é importantíssimo que seja realizado uma reinicialização no sistema, após tantas configurações, pois até o presente momento provisionamos um ambiente de computação em nuvem, incluindo sua instalação, rede e recursos do sistema operacional. Então para realizar a reinicialização do servidor basta executar o comando abaixo:

```
[root@master ~]# reboot
```

c) Criar uma Chave SSH para o Ambiente de Nuvem

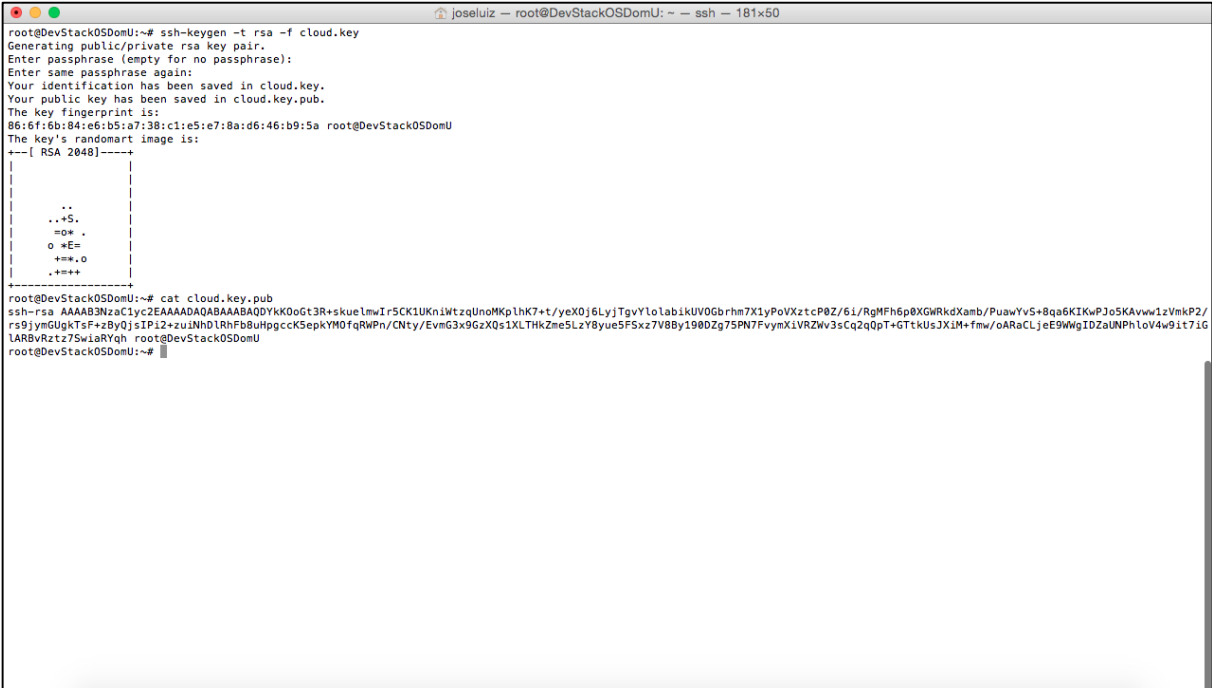
Por padrão o *Openstack Cloud* não possui nenhum sistema de autenticação para as máquinas virtuais. Permitindo assim que futuramente, configurar um servidor LDAP (Lightweight Directory Access Protocol) e configurar as imagens ou configurar em *Puppet* as configurações PAM necessárias para usar LDAP como método de autenticação as VMs, mas neste momento o *Openstack* permite que seja realizada após a instalação, visto que pode se houver uma chave de SSH configurada na VM ele pode realizar alguns procedimentos diretamente na VM como trocar uma senha de um usuário específico após a instalação e a forma mais simples é utilizar o serviço

de metadados, que somente serão funcionais após importar a chave pública no *Openstack* e atribuí-la a uma VM convidada, conforme procedimentos abaixo:

```
[root@master ~]# ssh-keygen -t rsa -f cloud.key
```

Na Figura 13, podemos visualizar a saída do comando informado acima, assim como o conteúdo da chave criada, que será utilizado na próxima etapa:

Figura 13 – Geração de Chaves de Segurança para Instâncias do *Openstack*



```

root@DevStackOSDomU:~# ssh-keygen -t rsa -f cloud.key
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in cloud.key.
Your public key has been saved in cloud.key.pub.
The key fingerprint is:
86:6f:6b:84:e6:b5:a7:38:c1:e5:e7:8a:d6:46:b9:5a root@DevStackOSDomU
The key's randomart image is:
+--[ RSA 2048 ]-----+
|
|
|
|
|
|
|
|
|
|
+-----+
root@DevStackOSDomU:~# cat cloud.key.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDkK0Gt3R+skuelmwIr5CK1UKniWtzqUn0MKpLhk7+t/yeX0J6lyjTgvY1oLabkIUV0Gbrhm7X1yPoVXztcP0Z/61/RgMFh6p0XGWRkdXamb/PuawYvS+8qa6KIKwP3o5KAvww1zVmkP2/
rs9jymGUgkTsF+zByDjsIP12+zuzINHdLRhFbBuHggccK5epkYMOtqRwPh/NTy/EvmG3x9GzX0s1XLThkZme5LzY8yue5F5sz7V8BY1980Zg75PN7FvymX1VRZwv3sCq2q0pT+GTtkUsJX1M+fmw/oARaCLjeE9WgIDZaUNPhLoV4w9it7iG
LARbVrtz7SwlaRYqh root@DevStackOSDomU
root@DevStackOSDomU:~#

```

Fonte: Desenvolvido pelo autor.

Este procedimento irá facilitar o uso de senhas, visto que estamos utilizando um ambiente controlado, assim não necessitando que seja digitado uma senha a todo momento.

Após a criação da chave SSH, você deseja iniciar uma VM que podemos usar para importar a nossa chave SSH. Entre para o painel *Dashboard*¹ (Horizon) novamente usando as credenciais fornecidas *keystonerc_admin*. Acesse >> Projeto >> computação >> Instâncias e clique no botão "Disparar Instância", no canto superior

¹ Através do link: <http://192.168.10.2/dashboard>

direito. Você será apresentado com um formulário conforme imagem abaixo e após clique em “Disparar” no canto inferior direito:

Figura 14 – Tela de Criação de uma Instância

192.168.10.102

Disparar Instância

Detalhes * Acesso e Segurança * Pós-Criação Advanced Options

Zona de Disponibilidade
nova

Nome da instância *
web01.maestrohost.com.br

Flavor *
m1.medium

Contagem de Instâncias *
1

Origem da Inicialização da Instância *
Boot from image (creates a new volume)

Nome da Imagem
cirros-0.3.2-x86_64-uec (24,0 MB)

Tamanho do Dispositivo (GB)
1

Nome do Dispositivo
vda

Excluir ao Finalizar

Especifique os detalhes para disparar uma instância.
O gráfico abaixo mostra os recursos usados por este projeto em relação às cotas do projeto.

Detalhes do Flavor

Nome	m1.medium
vCPUs	2
Disco Raiz	40 GB
Disco Temporário	0 GB
Disco Total	40 GB
RAM	4,096 MB

Limites de Projeto

Número de Instâncias 0 de 10 Utilizado

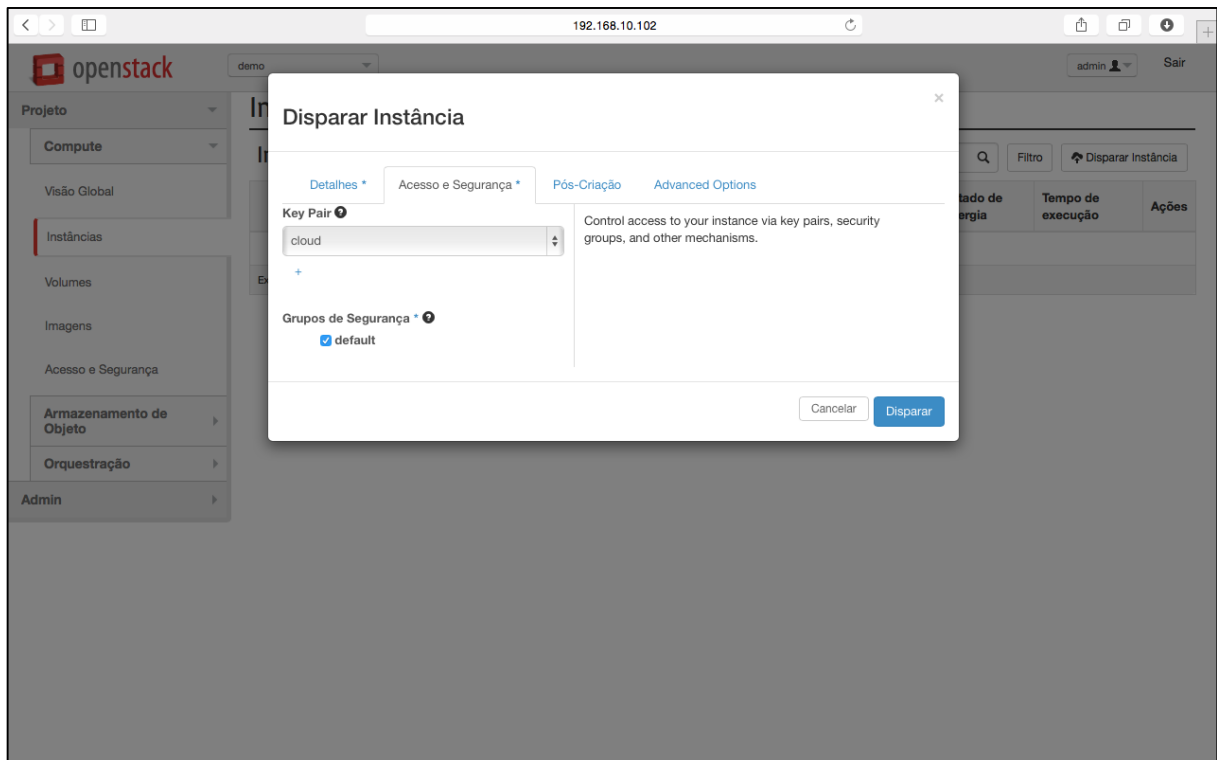
Número de vCPUs 0 de 20 Utilizado

RAM Total 0 de 51.200 MB Utilizados

Cancelar Disparar

Fonte: Desenvolvido pelo autor.

Figura 15 – Tela de Seleção de Chave de Segurança para uma Instância



Fonte: Desenvolvido pelo autor.

Figura 16 – Tela de Inicialização de uma Instância

The screenshot shows the OpenStack dashboard interface. On the left, there is a navigation menu with categories like 'Projeto', 'Compute', 'Visão Global', 'Instâncias', 'Volumes', 'Imagens', 'Acesso e Segurança', 'Armazenamento de Objeto', 'Orquestração', and 'Admin'. The main content area is titled 'Instâncias' and contains a table of instances. A single instance is listed with the following details:

Nome da instância	Nome da Imagem	Endereço IP	Tamanho	Key Pair	Status	Zona de Disponibilidade	Tarefa	Estado de energia	Tempo de execução	Ações
<input type="checkbox"/> web01.maestrohost.com.br	-		m1.medium	cloud	Build	nova	Block Device Mapping	No State	0 minuto	Associar IP Flutuante Mais ▾

Below the table, it indicates 'Exibindo 1 item'. In the top right corner, a green notification box displays the message: 'Sucesso: Disparadas instância chamadas "web01.maestrohost.com.br".' Above the table, there are buttons for 'Disparar Instância', 'Soft Reboot Instâncias', and 'Terminar Instâncias'.

Fonte: Desenvolvido pelo autor.

4 RESULTADOS E DISCUSSÃO

Este capítulo dedica-se a expor e comparar as duas maneiras de entregar serviços de TI. Será dividido em 2 subcapítulos, que abordarão os quatro objetivos desta pesquisa, que são:

a) Demonstrar que através deste método proposto na pesquisa reduz-se o tempo de provisionamento de recursos como o armazenamento, conectividade, plataformas, aplicações e serviços, escalável e de alta disponibilidade.

b) Comparar o tempo de uma implantação com e sem este método.

c) Separar o conceito de arquiteto de soluções de TI do responsável pela instalação.

d) Produzir uma máquina virtual (*appliance*) instalação de um Sistema Operacional de base Linux para a fácil e rápida instalação de uma infraestrutura de computação em nuvem.

4.1 RESULTADO COMPARATIVO ENTRE O MÉTODO TRADICIONAL E O PROPOSTO SOBRE ENTREGA DE SERVIÇOS DE INFRAESTRUTURA DE TI

Atualmente o modelo utilizado para provisionamento de infraestrutura para fornecer serviços de TI é manual, de forma que ao surgir a demanda por parte do cliente, deve-se efetuar uma solicitação que terá sua possibilidade analisada. Em caso de haver a possibilidade, o hardware deve ser alocado em algum espaço físico. Após a instalação física, inicia-se a instalação lógica, a qual contempla a escolha do sistema e instalação do sistema operacional e software que serão utilizados e disponibilizados como serviço. Todo este processo demora, normalmente de 1 a 3 dias, principalmente se não houver disponibilidade de recursos computacionais. Muitas vezes, estes recursos não estão disponíveis por já estarem alocados a outros projetos e/ou clientes, porém, com recursos ociosos.

O método tradicional pode ser entendido como a implantação de um servidor físico dedicado, ou seja não se trata de um ambiente compartilhado e o sistema operacional tem todos os recursos à sua disposição. É menos escalável que um servidor virtual, no entanto é mais confiável em termos de desempenho, pois o sistema não compartilha os recursos de hardware com mais ninguém. Os seus recursos são

um todo, e não uma parte de um todo. Este modelo de uso é mais caro, uma vez que envolve mais hardware, e o sistema tem acesso a todos os recursos, mesmo que estes não estejam em uso (PPLWARE, 2014).

O método proposto é o IaaS, isto é, uma plataforma de computação em nuvem que é o mais recente e poderoso modelo de alocação de servidores, pois consegue juntar as vantagens dos dois modelos, a virtualização e o ambiente de servidores físicos. Este tipo de alocação permite o acesso a um pool de recursos computacionais conforme as necessidades reais de cada sistema (*on-demand*). O próprio usuário pode provisionar os recursos que deseja como o CPU, disco, memória RAM e largura de banda, sem a necessidade de interação com o provedor do serviço. A escalabilidade dos recursos é fácil e rápida e o utilizador só paga pelos recursos que consumiu. A melhor característica deste modelo, é a flexibilidade que ele oferece, uma vez que o usuário define exatamente o que deseja e quando deseja. Tudo isto, em minutos. É possível controlar os ambientes computacionais em um modelo *pay-per-use* que é entregue como um serviço, evitando assim mensalidades, anuidades e contratos fixos, pagando apenas pelo uso, baseados no uso real do serviço (PPLWARE, 2014).

Embora existam formas distintas de se solucionar o problema de tempo, ociosidade, escalabilidade e alta disponibilidade de recursos, todo e qualquer fornecedor de infraestrutura de TI deve-se atentar ao desenvolvimento de novos modelos (formas) de entregar serviço, pois a quantidade de demanda é diretamente proporcional a velocidade de entrega destes serviços, ao menos esta é a expectativa do cliente.

Abaixo está representado um comparativo entre os métodos tradicional e o método proposto, classificando os itens em Infraestrutura, Escalabilidade, Gerenciamento e Custo.

Figura 17 - Infraestrutura

Recursos	Método Tradicional	Método Proposto
Uso do Hardware	Um único servidor físico usado somente por um sistema operacional.	Sistema distribuído entre vários servidores físicos em um ambiente virtual isolado.

Otimização de Recursos		Depende diretamente dos recursos disponíveis no hardware.	Pode ser distribuído em diferentes nós do ambiente de computação em nuvem.
Provisionamento de Conectividade		Os serviços de rede pública estão configurados diretamente no sistema operacional.	Os serviços de rede estão divididos em rede de gerenciamento, público e privado sendo gerenciados por um firewall.
Provisionamento de Plataformas		Dependem de uma intervenção técnica para que a disponibilização da plataforma seja realizada.	Após definido um modelo pode-se automatizar através de uma imagem ou um <i>template</i> (modelo).
Provisionamento de Aplicações		Dependem de uma intervenção técnica para que a disponibilização de aplicações seja realizada.	Após definido um modelo pode-se automatizar através de uma imagem ou um <i>template</i> (modelo).
Provisionamento de Serviços		Dependem de uma intervenção técnica para que a disponibilização dos serviços seja realizada.	Após definido um modelo pode-se automatizar através de uma imagem ou um <i>template</i> (modelo).

Fonte: Desenvolvido pelo autor.

Como podemos observar na tabela acima quando falamos de uso de hardware, o método proposto tem destaque devido ao seu uso compartilhado do hardware, porém isolado; já o método tradicional pode utilizar todo o hardware somente para uma finalidade operacional o que traz menos benefícios para a operação. O provisionamento de conectividade, plataformas, aplicações e serviços após terem seus modelos e políticas de segurança, ou seja, uma arquitetura de soluções definidas pelo provedor de serviços para sua oferta, não necessitam de nenhuma intervenção técnica por parte do provedor. Este torna-se um benefício em comparação ao método tradicional que necessita quase que constantemente de uma intervenção técnica para provisionar toda conectividade, plataforma, aplicação ou serviço.

Figura 18 - Escalabilidade

Recursos	Método Tradicional	Método Proposto
Uso do Hardware	Não é possível upgrade ou downgrade. Caso sejam necessários mais recursos deve ser contratado um pacote diferente e previamente definido pelo provedor, ainda que os recursos não sejam os ideais.	Auto provisionamento e implantação simples e rápida (poucos cliques). Upgrade e downgrade em segundos, dimensionamento sem a interrupção dos serviços.
Otimização de Recursos	Depende diretamente dos recursos disponíveis no hardware.	Pode ser distribuído em diferentes nós do ambiente de computação em nuvem
Escalabilidade	Não há escalabilidade, visto que o hardware é exclusivo ao sistema.	A escalabilidade é fundamental para o ganho de desempenho quando necessário para uma aplicação.
Alta Disponibilidade	Ocorre através de recursos como o Heart beat. Necessita de outro hardware de mesma capacidade.	O ambiente sempre é virtualizado e um nó sempre utiliza o outro como contingência de disponibilidade e computação e armazenamento.

Fonte: Desenvolvido pelo autor.

Avaliando a tabela acima, no método tradicional, todo *upgrade* (acréscimo) ou *downgrade* (decréscimo) na configuração do hardware somente torna-se possível mediante a contratação de um novo pacote de serviço, o que provavelmente irá acarretar na sobra de recursos por parte do hardware, além de haver um *downtime*

(tempo de inatividade) para a realização de tal procedimento. No modelo proposto os recursos podem ser acrescentados ou retirados a qualquer momento sem a necessidade de intervenção técnica, disponibilizando os recursos de dimensionamento em segundos (PPLWARE, 2014).

A otimização dos recursos dependerá sempre dos recursos disponíveis no servidor físico, o que pode inviabilizar toda e qualquer otimização. Através do modelo proposto havendo a necessidade de otimização de recursos os mesmos podem ser distribuídos em diferentes nós de computação, o que pode ser entendido pelo usuário como recursos ilimitados.

A escalabilidade torna-se automática no modelo proposto, visto que havendo necessidade por parte do sistema o desempenho pode ser redimensionado atendendo assim a necessidade momentânea da aplicação, o que não pode ocorrer através do método tradicional devido ao seu sistema ser previamente alocado durante a instalação para um hardware em específico.

A alta disponibilidade pode ocorrer em ambos os ambientes, no método tradicional o mesmo pode ocorrer através do *Heartbeat*, ou seja pulsação, batimento cardíaco. O *Heartbeat* está para os servidores, assim como, o monitor cardíaco está para nós seres humanos. Mas além de monitorar os servidores, o *Heartbeat* é capaz de substituir um servidor por outro, caso o primeiro falhe, para manter o funcionamento do sistema (ENSINAR, 2008). Um malefício deste modelo é a necessidade de ter disponível um hardware equivalente e exclusivo ao que falhou para que seja utilizado somente quando houver falha. No método proposto podemos observar que através do ambiente virtualizado sempre há alta disponibilidade, pois um nó sempre pode ser contingência de outro em recursos computacionais como processador, memória e armazenamento.

Figura 19 - Gerenciamento

Recursos	Método Tradicional	Método Proposto
Gerenciamento Centralizado	Somente se utilizado um ambiente virtualizado e homogêneo.	O gerenciamento sempre é centralizado e gerenciado no servidor nó "master", através também do dashboard <i>Horizon</i> .

Gerenciamento de ambientes Heterogêneos	Não é possível o gerenciamento centralizado de ambientes heterogêneos.	De forma simplificada é possível gerenciar os nós de ambientes heterogêneos através do dashboard <i>Horizon</i> .
Processo de instalação Automatizado.	O processo de instalação é complexo e pode ser automatizado em etapas, mas nunca de forma completa.	O processo de instalação é complexo, mas pode ser automatizado.

Fonte: Desenvolvido pelo autor.

Conforme a tabela acima pode-se observar que o gerenciamento centralizado somente é possível se utilizado o método proposto, visto que utiliza de uma interface web para fácil gerenciamento, que é o *dashboard* denominado *Horizon* (ORACLE, 2014). O gerenciamento em ambientes físicos torna-se possível somente se o ambiente for convertido em virtual e homogêneo (CITRIX, 2011).

O gerenciamento de ambientes heterogêneos torna-se simples através do *Openstack*, visto que ele pode operar com qualquer *hypervisor*, então pode-se entender que gerencia ambientes heterogêneos da mesma forma, como se fossem ambientes homogêneos.

O processo de automatização de instalações pode se dar através de ferramentas como Ansible (RACKSPACE). Ansible é um mecanismo de automação de TI simples que automatiza o provisionamento em ambientes de nuvem, gerenciamento de configuração, implantação de aplicativos (ANSIBLE).

Figura 20 - Custo

Recursos	Método Tradicional	Método Proposto
Tempo de Provisionamento de recursos de Infraestrutura	Somente se houver recursos físicos disponíveis. Em média 3 dias.	A falta de recursos não é uma limitação, pois tendo o processo automatizado pode-se distribuir em diferentes nós que

		tenham tais recursos. Em média 8 horas.
Forma de faturamento	Pagamento é definido em pacotes previamente dimensionados, conforme oferta do provedor. Faturamento anual ou mensal, há necessidade de contrato.	Pagamento pay-per-use, ou seja paga-se por aquilo que se usa. Faturamento por hora de utilização, não há necessidade de contrato.

Fonte: Desenvolvido pelo autor.

Observa-se na tabela acima que o tempo de provisionamento para uma infraestrutura física, dedicada e exclusiva depende de recursos físicos necessários e pode levar em média 3 dias para ser disponibilizado para o cliente (LOCAWEB, [2013]). Levando em consideração que o tempo de inatividade tem um custo para o negócio e que tudo atualmente acontece muito rápido, é considerado um tempo elevado e entendido como um custo. No método proposto a falta de recursos praticamente não existe, pois tendo o serviço automatizado a instância (máquina virtual) pode ser carregada em qualquer nó do ambiente de nuvem que tenha os recursos computacionais necessários para carregar esta instância, conforme apresentado na metodologia deste trabalho.

O faturamento dos clientes no método tradicional se dá através da contratação de pacotes de serviços denominados planos e sempre vinculados a um contrato (LOCAWEB 2, [2013]). Os planos muitas vezes não são exatamente o que é necessário para a execução da aplicação, sendo assim pode-se passar parte do tempo sem utilizar todos os recursos do hardware, assim como pode faltar quando necessário. No modelo proposto o faturamento se dá através da contabilização do uso dos recursos computacionais e armazenamento baseado em horas, ou seja pago somente o que consumir e somente se consumir, conhecido como modelo *pay-per-use*, normalmente sem a necessidade de contratos (CLOUDD8, [20--]).

4.2 SEPARAÇÃO DA INTELIGÊNCIA DA ARQUITETURA E DA IMPLANTAÇÃO

Durante a implantação do serviço de computação em nuvem é possível observar que a implantação dos serviços contratados não depende de uma intervenção técnica de alta capacidade. Esta intervenção é realizada no momento do desenvolvimento da arquitetura do ambiente de computação em nuvem, fazendo assim com que sejam separadas as atividades técnicas operacionais das atividades de arquitetura e projeto.

A separação da inteligência de arquitetar a solução e da implantação da solução se dá devido a necessidade de responder possíveis problemas que venham surgir durante o processo de implantação e, sendo o arquiteto de soluções e o técnico dois profissionais diferentes, tem-se a possibilidade de uma solução mais precisa e rápida.

Tomando como base este conceito de separação pode-se entender o envolvimento e o conhecimento técnico aplicado diretamente a parte operacional e o arquiteto diretamente aplicado ao negócio. Este conceito torna-se um ganho para as empresas que conseguem evoluir em uma escala maior, com uma mão de obra especificamente qualificada e focada em suas respectivas atividades.

4.3 RESULTADOS DA PRODUÇÃO DE UMA MÁQUINA VIRTUAL (APPLIANCE) COM OPENSTACK

A produção de uma máquina virtual que será chamada de *appliance* tem o objetivo de automatizar a entrega de uma plataforma de computação em nuvem utilizando a ferramenta *Openstack*. A utilização do sistema operacional Linux foi de extrema importância, visto que para a implantação deste ambiente não foi necessário nenhum investimento em licenciamento de software, assim a escolha se deu pela distribuição Ubuntu, mais especificamente, o Ubuntu Server 64 bits 14.04.1 LTS devido a sua particular segurança e avanço nas atualizações de softwares que complementam o sistema operacional.

Visando facilitar a implantação e a entrega dos serviços por parte dos provedores de serviços de TI, esta *appliance* traz uma capacidade de conhecimento de diversas áreas que envolvem a infraestrutura de TI, como redes de computadores, segurança, armazenamento, virtualização e sistemas operacionais.

Como rede, o projeto Neutron tem como objetivo certificar que as instâncias estejam configuradas de tal forma que possam se comunicar entre si e com o mundo

exterior. Para isso, as instâncias têm dois endereços IP associados, um para cada caso. Em outros termos é a conectividade de rede como serviço.

Como computação, o Nova fornece todos os elementos necessários para dar suporte aos serviços de computação virtual. Criando uma plataforma escalável e redundante que compreende grandes redes e recursos de servidores. O Nova também disponibiliza a interface de gerenciamento da nuvem, o *Horizon*, isso inclui o gerenciamento de instâncias da nuvem, assim como seus recursos computacionais como processador, memória e armazenamento.

Como gerenciador de imagens, o *Glance* é basicamente um sistema de registro e recuperação de imagens em servidores. Permite também que essas imagens sejam catalogadas e utilizadas sempre que necessário, isso facilita o reuso de uma instalação e tornando desnecessário o retrabalho de um profissional após a instalação.

Como gerenciamento de armazenamento, o *Swift* ou *Object Storage* teve como base o projeto *Cloud Files* que foi implementado pela *Rackspace* e trouxe um ganho inigualável para o gerenciamento de arquivos para a mesma. Este é o responsável por fornecer serviço de armazenamento dentro do *Openstack*. O *Object Storage* faz uso de uma arquitetura distribuída e os objetos são gravados em vários dispositivos de armazenamento, ou seja, em diversos locais remotos, criando assim uma alta capacidade de redundância ou *cluster*. Neste formato que ele garante a integridade dos dados é muito similar ao serviço S3 da Amazon (UFRJ, 2013).

Esta *appliance* proporciona uma forma simples e objetiva de entregar todos estes conhecimentos embarcados em um serviço.

5 CONSIDERAÇÕES FINAIS

Os objetivos deste trabalho foram: (a) apresentar o modelo de entrega de serviços de TI denominado computação em nuvem; (b) um método de implantação de uma infraestrutura de computação em nuvem; (c) um comparativo entre a implantação do método tradicional e o método proposto; (d) a separação da inteligência de uma arquitetura e de sua implantação; assim como (e) o desenvolvimento de uma máquina virtual com a ferramenta *Openstack* e todos os recursos descritos neste trabalho para uma rápida e fácil implantação.

Através desta pesquisa foi possível concluir que o modelo denominado computação em nuvem é uma evolução da forma de se entregar serviços de TI através da internet, o que vem revolucionando o mercado e acelerando cada vez mais as demandas e conseqüentemente suas entregas.

O método proposto foi a utilização dos recursos de virtualização, o que é toda a base da disseminação do modelo de computação em nuvem, o desenvolvimento de modelos de imagens de sistemas operacionais para a rápida disponibilização de recursos de processamento, armazenamento e escalabilidade para os clientes que desses recursos necessitem. A disponibilização desta plataforma tem como objetivo reduzir o tempo de entrega de serviços de infraestrutura como um serviço para clientes utilizando a ferramenta *Openstack*.

A ferramenta *Openstack*, selecionada para a implantação do ambiente aqui descrito, se demonstrou estável e fiel em todos os momentos que utilizou os recursos disponíveis em sua plataforma, confirmando-se uma ótima opção além de ser de código aberto. Foi possível observar que diversos métodos de implantação desta ferramenta estão disponíveis e que cada vez mais os mesmos vem facilitar a implantação de ambientes de alta disponibilidade com o *Openstack*.

Através de um comparativo desenvolvido e apresentado neste trabalho foi possível observar que o uso do método proposto tem grandes benefícios para o fornecimento de serviços de TI, principalmente em infraestrutura de TI que é o foco deste trabalho. O tempo de disponibilização de uma plataforma é reduzido drasticamente e por sua conseqüência a disponibilização de recursos de hardware como processamento e armazenamento também, pois uma vez desenvolvido o modelo e definidos planos para disponibilizar estes serviços para o mercado, somente

será necessário o usuário acessar e gerenciar seus recursos, sempre através do *dashboard*, sem a necessidade de nenhuma intervenção técnica para o acréscimo ou decréscimo destes recursos.

A abordagem da separação da arquitetura de um ambiente de nuvem de sua implantação deu-se de forma a objetivar a ocupação de técnicos de alta capacidade para alinhamento com o negócio como arquitetar o ambiente, tornando assim desnecessário seu envolvimento de intervenção no momento da implantação, e formando especialistas em novas áreas, como para a implantação destes ambientes arquitetados.

A produção de uma máquina virtual de base Linux com o sistema *Openstack* instalado traz a facilidade no momento da implantação, sem a necessidade de se planejar um ambiente de nuvem sempre que necessário e, sim, dimensionar o tamanho dos recursos que se quer disponibilizar para a arquitetura desta plataforma. Esta máquina virtual servirá como modelo para a comercialização dos serviços da empresa Horizon IT, de propriedade do autor, com o objetivo de contribuir com o desenvolvimento das infraestruturas de seus clientes, tornando-os cada vez mais preparados para as altas e constantes demandas de seus clientes. Dessa forma, o autor acredita ter contribuído com a comunidade tecnológica, pois respondeu todos os itens a qual essa pesquisa se propõem.

REFERÊNCIAS

ADRENALINE. **O que é um NAS (Network Attached Storage)?**. Disponível em: <<http://adrenaline.uol.com.br/tecnologia/artigos/230/o-que-e-um-nas-network-attached-storage.html>>. Acesso em: 01 out. 2014.

AFAZEKAS. Getting Started With XenServer and Devstack. [S.l.]: GitHub, [2014]. Disponível em: <https://github.com/openstack-dev/devstack/blob/master/tools/xen/README.md>>. Acesso em 16 nov. 2014.

ANSIBLE. **WHAT IS ANSIBLE**, 2014, Disponível em: <<http://www.ansible.com/how-ansible-works>>. Acesso em: 20 nov. 2014.

ARMBRUST, Michael et al. **Above the Clouds: A Berkeley View of Cloud Computing**. Electrical Engineering and Computer Sciences University of California at Berkeley. Technical Report No. UCB/EECS-2009-28. Disponível em: <<http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>>. Acesso em: 01 out. 2014.

BURGER, Thomas. **The Advantages of Using Virtualization Technology in the Enterprise**. Disponível em: <<http://software.intel.com/en-us/articles/the-advantages-of-using-virtualization-technology-in-the-enterprise/>>. Acesso em: 29 out. 2014.

CAMPOS, F. F. de. **O que é Heartbeat?**. [S.l.]: Ensinar, 2008. Disponível em: <<http://ensinar.wordpress.com/2008/12/01/o-que-e-heartbeat/>>. Acesso em 16. nov. 2014.

CENTOS. **CentOS Linux**. Disponível em: <[http:// https://www.centos.org/about/](http://https://www.centos.org/about/)>. Acesso em: 02 jul. 2014.

CITRIX, **Citrix XenServer**: Ferramentas de virtualização, automatização e gerenciamento avançado para o datacenter. 2011. Disponível em: <http://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/citrix-xenserver-product-overview-br.pdf>. Acesso em: 02 nov. 2014.

CLOUD8. **Modelo de custo**: pay per use. [20--]Disponível em: <<https://www.cloud8.com.br/ajuda/conceitos/modelo-de-custo-pay-per-use/>>. Acesso em: 20 nov. 2014.

DEVSTACK, Core Team. **DevStack**. 2014. Disponível em: <<https://wiki.openstack.org/wiki/DevStack>>. Acesso em: 16 nov. 2014.

DUALTEC. **Entendendo as camadas do cloud computing**. Disponível em: <http://lt.idg.com.br/Dualtec/Dualtec_whitepaper_as_camadas_da_nuvem.pdf>. Acesso em: 02 nov. 2014.

FLÁVIO, R. C. et al. Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios. In: ERCEMAPI, 2010. Disponível em: <https://www.academia.edu/783784/Computa%C3%A7%C3%A3o_em_Nuvem_Con>

ceitos_Tecnologias_Aplica%C3%A7%C3%B5es_e_Desafios>. Acesso em: 20 nov. 2014.

FOSTER I. et al. "**Cloud Computing and Grid Computing 360-Degree Compared**" – *Department of Computer Science, University of Chigado*, 2008.

GOOGLE TRENDS. **Openstack, Cloudstack e Amazon AWS**. Disponível em <<https://www.google.com.br/trends/explore#q=openstack%2C%20Cloudstack%2C%20Amazon%20AWS&cmpt=q>>. Acesso em: 27 out. 2014.

GOOGLE TRENDS. **Servidor DELL, Servidor IBM, Servidor HP e IaaS**. Disponível em: <<https://www.google.com.br/trends/explore#q=Servidor%20DELL%2C%20Servidor%20IBM%2C%20Servidor%20HP%2C%20IaaS&cmpt=q>>. Acesso em: 28 out. 2014.

IBM. **Computação em Nuvem e Armazenamento com OpenStack**. Disponível em: <<http://www.ibm.com/developerworks/br/cloud/library/cl-openstack-cloud/>>. Acesso em: 28 out. 2014.

IBM. **OpenStack: Computação em Nuvem Flexível e Aberta**. Disponível em: <<https://www.ibm.com/developerworks/community/blogs/tlcb/entry/mp219?lang=en>>. Acesso em: 28 out. 2014.

IEEE. **Cloud Computing**. Disponível em: <<https://wiki.openstack.org/wiki/XenServer>>. Acesso em: 05 out. 2013.

INFOWESTER. **O que é Linux e Qual a sua História?**. [S.l.]: Infowester, 2011. Disponível em: <http://www.infowester.com/historia_linux.php>. Acesso em 05 nov. 2014.

LOCAWEB. **Sobre os Servidores Dedicados**. [2013]. Disponível em: <<http://www.locaweb.com.br/produtos/servidores-dedicados/duvidas-frequentes.html>>. Acesso em: 20 nov. 2014.

LOCAWEB 2. **Planos Servidores Gerenciados**. [2013]. Disponível em: <<http://www.locaweb.com.br/produtos/servidores-dedicados/planos-gerenciados.html>>. Acesso em: 20 nov. 2014.

MELL, Peter; GRANCE, Timothy. *The NIST definition of cloud computing (draft)*. NIST special publication, v. 800-145, p. 2, 2011.

NIST. **The NIST Definition of Cloud Computing**. Special Publication 800-145. 2011. Disponível em: <<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>>. Acesso em: 16 nov. 2014.

OPENSOURCE. **What is Openstack?**. [2008]. Disponível em: <<http://opensource.com/resources/what-is-openstack>>. Acesso em: 05 out. 2014.

OPENSTACK. **Xen, XenAPI, XenServer and XCP**. Disponível em: <<http://docs.openstack.org/grizzly/openstack-compute/admin/content/introduction-to-xen.html>>. Acesso em: 05 out. 2013.

OPENSTACK. **XenServer**. Disponível em: <<https://wiki.openstack.org/wiki/XenServer>>. Acesso em: 05 out. 2013.

OPENSTACK. **XenServer**. Disponível em: <<https://http://cloudcomputing.ieee.org/education#elearning>>. Acesso em: 02 jul. 2014.

ORACLE, Gerenciamento Centralizado de Nuvem com OpenStack, 2014. Disponível em <https://docs.oracle.com/cd/E56346_01/html/E53697/golpl.html>. Acesso em 20/11/2014.

PPLWARE, **VPS vs Servidor Dedicado vs Servidor Cloud, 2014**. Disponível em: <<http://pplware.sapo.pt/informacao/vps-vs-servidor-dedicado-vs-servidor-cloud/>>. Acesso em: 19 nov. 2014.

RACKSPACE. **OpenStack**: a alternativa aberta para restringir a nuvem. Disponível em: <<http://www.rackspace.com/pt/cloud/openstack/>>. Acesso em: 05 out. 2013.

RACKSPACE, Rackspace Support. **Understanding the Cloud Computing Stack: SaaS, PaaS, IaaS**. 2013. Disponível em: <http://www.rackspace.com/knowledge_center/whitepaper/understanding-the-cloud-computing-stack-saas-paas-iaas>. Acesso em: 02 nov. 2014.

REDHAT. **RDO Home Page**. Disponível em: <https://openstack.redhat.com/Main_Page>. Acesso em: 20 out. 2014.

SLIDESHARE. **O Futuro do Openstack, NFV, SDN e Cloud Computing**. Disponível em: <<http://pt.slideshare.net/mdieder/o-futuro-do-openstack-nfv-sdn-e-cloud-computing>>. Acesso em: 27 out. 2014.

SOFTWARE LIVRE. **OpenStack** – Um sistema operacional na nuvem de código aberto. 2013. Disponível em: <<http://softwarelivre.org/georgemendonca/blog/openstack-%E2%80%93-um-sistema-operacional-na-nuvem-de-codigo-aberto>>. Acesso em: 01 nov. 2014.

SOUSA, Flávio R. C et al. **Computação em Nuvem: Conceitos, Tecnologias, Aplicações e Desafios**. Teresina: ERCEMAPI, 2009.

TORVALDS, Linus; DIAMOND, David. **Só por prazer: Linux, os bastidores da sua criação**. Rio de Janeiro: Campus, 2001.

UFRJ. **Openstack**. 2013. Disponível em: <http://www.gta.ufrj.br/grad/13_1/openstack/>. Acesso em: 16 nov. 2014.

UFRJ. **Openstack**, 2013. Disponível em: <http://www.gta.ufrj.br/grad/13_1/openstack/>. Acessado em: 18 nov. 2014.

VANOVER, R. **FreeNAS for beginners: What it can and cannot do**. [S.l.]: Data center, 2011. Disponível em: <<http://www.techrepublic.com/blog/data-center/freenas-for-beginners-what-it-can-and-cannot-do/>>. Acesso em: 20. nov. 2014.

UFRJ. **O que é o hypervisor ou VMM (Virtual Machine Monitor - Monitor de Máquina Virtual)?**. Rio de Janeiro: UFRJ, 2008. Disponível em:

<[http://www.gta.ufrj.br/grad/08_1/virtual/OqueohypervisorouVMM\(VirtualMachineMonit.html](http://www.gta.ufrj.br/grad/08_1/virtual/OqueohypervisorouVMM(VirtualMachineMonit.html)>. Acesso em 20 nov. 2014.

VECCHIOLA, C.; CHU, X.; BUYYA, R. **Aneka**: A Software Platform for .NET-based Cloud Computing, pages 267–295. In: W. Gentsch, L. Grandinetti, G. Joubert (Eds.). High Speed and Large Scale Scientific Computing. IOS Press, Amsterdam, Netherlands, 2009.

VELTE, Anthony T.; VELTE, Toby J.; ELSENPETER, Robert. **Cloud Computing: A Practical Approach**. McGraw-Hill, Inc., 2009.

VON HAGEN, William. **Professional Xen Virtualization**. 2008. Disponível em: <<http://e-books.bassq.nl/Citrix/Xen%20Virtualization%202008.pdf>>. Acesso em: 23 abr. 2011.

WANG, L., LASZEWSKI G. V. **Cloud Computing: a Perspective Study - Service Oriented Cyberinfrastructure Lab**, Rochester Inst. of Tech, 2008.