Capítulo

4

Análise do tráfego de máquinas virtuais na rede de controle de nuvens computacionais baseadas em OpenStack

Charles Christian Miers (UDESC), Guilherme Piêgas Koslovski(UDESC), Maurício Aronne Pillon(UDESC), Adnei Willian Donatti(UDESC)

Resumo

A computação em nuvem é um paradigma amplamente difundido para disponibilização de recursos computacionais sob demanda. Como suporte para o gerenciamento e provisionamento dos serviços ofertados, emprega-se a virtualização dos recursos de processamento, armazenamento e comunicação. Assim, o usuário pode ter acesso a uma infraestrutura virtualizada e privada mediante pagamento pelo uso, ou com a implantação de uma nuvem privada, cuja utilização alinha-se às necessidades de uma organização. A literatura especializada aponta que questões relacionadas ao desempenho de uma nuvem dependem da identificação de comportamentos e operações que ocorrem durante o seu uso. Além disso, percebe-se, que há uma necessidade de identificar operações, muitas das quais ocorrem no nível da camada de rede e demandam análise e compreensão do que está trafegando e sua finalidade. Neste sentido, a caracterização de tráfego auxilia este entendimento, por meio do emprego de técnicas e métodos que possibilitam a coleta e identificação de forma sistematizada. O presente minicurso apresenta a análise do tráfego de máquinas virtuais na rede de controle de nuvens computacionais gerenciadas por OpenStack. Além da descrição e análise do cenário, o minicurso apresenta resultados experimentais obtidos em uma nuvem computacional privada.

4.1. Conceitos básicos

A computação em nuvem (Subseção 4.1.1) possibilita uma forma otimizada e sob demanda de fornecer e consumir recursos computacionais como processamento, armazenamento e rede. Neste contexto, as soluções de nuvem empregam mecanismos de orquestração e gerenciamento para que diversas tecnologias já existentes possam ser empregadas harmoniosamente. Assim, soluções de nuvem empregam sistemas operacionais, técnicas de virtualização e sistemas de arquivos que já existem há décadas mas organizados

e gerenciados de uma forma diferente. A virtualização é um dos principais conceitos, não apenas através das máquinas virtuais (MVs) (Subseção 4.1.2) mas também através de recursos de rede Subseção 4.1.3).

4.1.1. Computação em Nuvem

Existem várias definições sobre computação em nuvem, contudo, devido a ampla adoção, a que mais se aproxima de uma padronização é a definição proposta pelo *National Institute of Standards and Technology* (NIST) [Mell and Grance 2011], que a trata como um modelo de computação que possibilita acesso a um conjunto de recursos computacionais (*e.g.*, armazenamento, rede e serviços) de maneira conveniente, ubíqua e escalável. Ainda, de acordo com o NIST, a oferta do serviço de computação em nuvem pode ser classificada em três categorias: *Infraestructure as a Service* (IaaS), *Platform as a Service* (PaaS) e *Software as a Service* (SaaS). O IaaS permite que o consumidor tenha acesso a recursos de mais baixo nível, como rede, armazenamento e processamento. Neste sentido, o consumidor pode até mesmo controlar o software na MV. Já no PaaS, o consumidor tem acesso a uma plataforma pré-configurada com um conjunto de linguagens e bibliotecas. Neste caso, o consumidor pode controlar as aplicações que serão instaladas por ele, bem como algumas configurações sobre o ambiente. Em relação ao modelo SaaS, as permissões concedidas ao consumidor são de mais alto nível. Então, há acesso às aplicações executando sobre a infraestrutura da nuvem, bem como a algumas configurações específicas para aquela aplicação. É importante ressaltar que variações e especificações das categorias foram propostas pelos provedores de nuvem, *e.g.*, *Function as a Service* (FaaS), *Container as a Service* (CaaS), *Load Balance as a Service* (LBaaS), entre outros.

O NIST também classifica as nuvens computacionais de acordo com seu modelo de implantação, sendo os modelos público e privado os mais comuns. Na nuvem pública a infraestrutura é de uso aberto, e é gerenciada por alguma entidade (*e.g.*, empresa, organização governamental, organização acadêmica). Já as nuvens computacionais privadas operam sobre uma infraestrutura própria, que é mantida pela organização que a possui, *i.e.*, toda a manutenção da nuvem, bem como aspectos de segurança e desempenho são de responsabilidade desta organização. Além disso, as nuvens privadas buscam atender aos propósitos da organização, e são acessíveis somente aos indivíduos com permissão, o que garante à organização total controle sobre os dados ali contidos [Jadeja and Modi 2012]. Dentre os softwares de código aberto que permitem a criação de nuvens públicas e privadas, destacam-se o OpenStack [OpenStack 2019b] e o CloudStack [project 2019]. Sendo o OpenStack a solução de código aberto IaaS mais empregada mundialmente.

Os principais atores envolvidos no modelo de computação em nuvem são: consumidor, que é o utilizador dos serviços da nuvem; provedor, que é a parte responsável por disponibilizar o serviço ao consumidor; auditor, que é responsável por avaliar a nuvem de modo geral (*e.g.,* avaliações de segurança e desempenho); corretor, cuja aplicabilidade se dá no controle de uso, desempenho e distribuição de serviços em nuvem; e operador, que atua como um intermediário entre o provedor e consumidor, de modo a possibilitar a conectividade e transporte de serviços entre eles [Bohn et al. 2011]. A Figura 4.1 ilustra o modelo de referência para computação em nuvem de acordo com o NIST. Especificamente, a Figura 4.2 apresenta o relacionamento entre alguns desses principais atores. Os usuários dos serviços solicitam os recursos para um provedor de infraestrutura. Cada

serviço é oferecido e executado sobre um conjunto de servidores virtualizados.

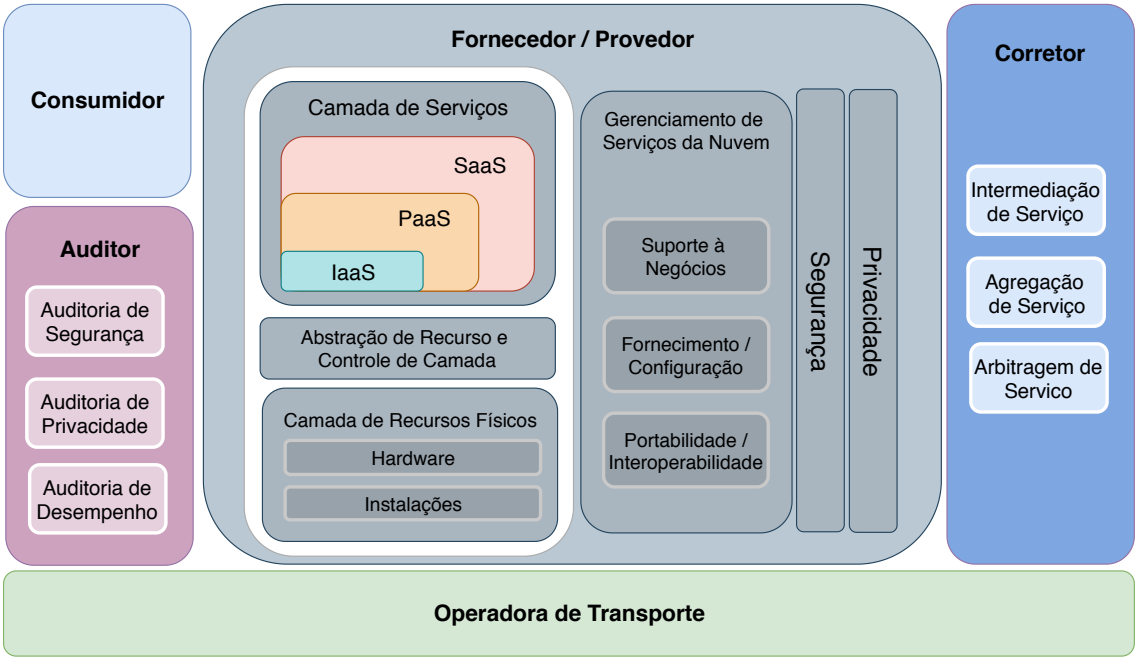

Figura 4.1: Modelo de referência para computação em nuvem de acordo com o NIST [Bohn et al. 2011].

Uma das premissas deste modelo de computação é que vários usuários possam solicitar os seus serviços simultaneamente. Dessa forma, todos os usuários fazem uso da mesma infraestrutura da nuvem (Figura 4.2), que por sua vez, é composta por elementos como servidores, roteadores, *switches* e infraestrutura (*e.g.*, *Ethernet*, fibra ótica e coaxial), que podem causar gargalos que afetam negativamente o fornecimento do serviço.

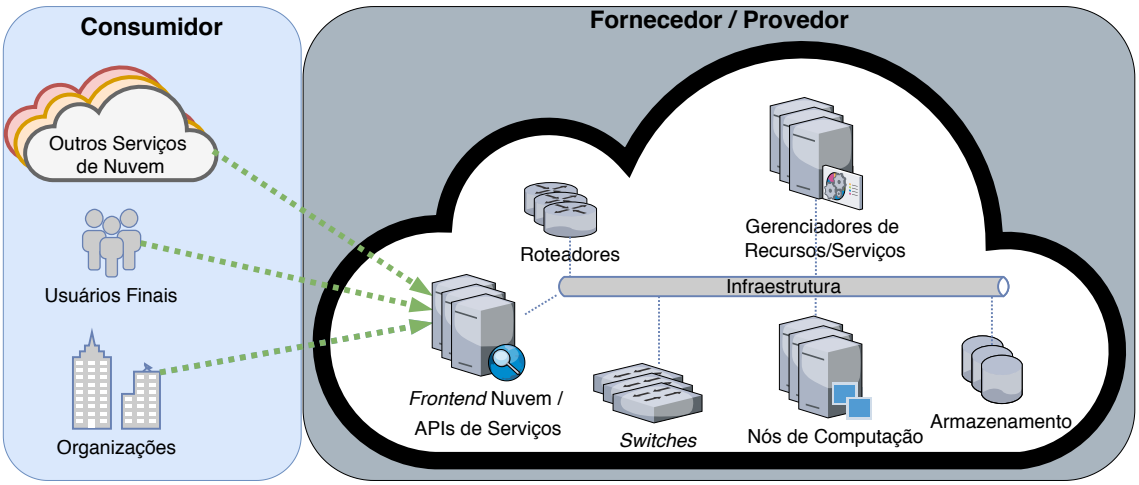

Figura 4.2: Principais atores do modelo de computação em nuvem.

Portanto, independente do modelo, cenário e solução de gerenciamento utilizado, o monitoramento e conhecimento sobre as particularidades da carga computacional e de comunicação são essenciais para garantir qualidade na oferta de serviço e no gerenciamento da nuvem (*e.g.*, configurações de rede e alocação de recursos).

4.1.2. Virtualização de Recursos Computacionais

De acordo com o NIST, a virtualização é a simulação do *software* e/ou *hardware* no qual outro *software* é executado. Além disso, existem diversas formas de virtualização, inclusive, não apenas máquinas virtuais (MVs) podem ser criadas, como contêineres, *virtual appliances* e também na parte de infraestrutura com redes virtualizadas (roteadores, *switches* e VLANs). Neste sentido, um dos principais benefícios para a crescente adoção da virtualização é a eficiência operacional: as organizações conseguem aumentar a carga de trabalho por hardware, uma vez que um único computador pode hospedar diversas máquinas virtuais simultaneamente [Scarfone et al. 2011]. Existem diferentes formas de fornecer recursos virtualizados. Porém, quase toda virtualização tem por objetivo propiciar alguma abstração para um serviço ou uma aplicação. Existem diferentes abordagens para fornecer os recursos computacionais para uma aplicação (Figura 4.3).

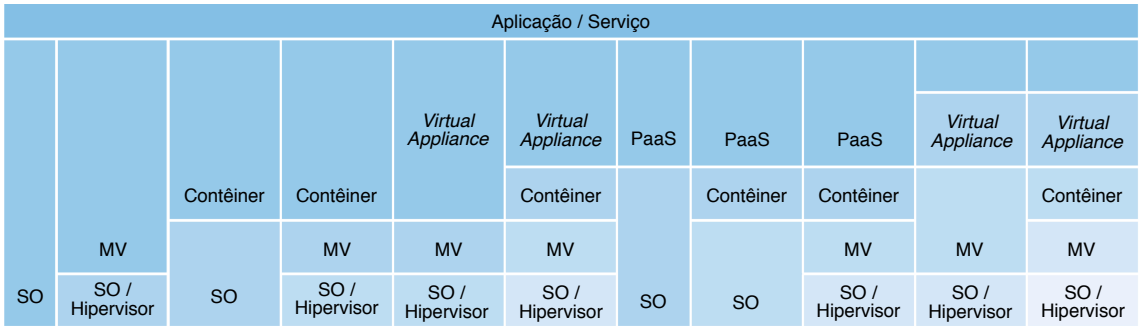

Figura 4.3: Principais formas de prover recursos computacionais a uma aplicação [Panizzon et al. 2019].

Observando a Figura 4.3, constata-se que pode-se usar desde uma abordagem mais antiga com apenas o SO – Aplicação/Serviço até abordagens com SO/Hipervisor – MV – Contêiner – *Virtual Appliance* – Aplicação/Serviço. Apesar de existirem estas diversas combinações, e a adoção de contêineres estar crescendo consideravelmente, o uso de MVs continua muito popular quer na configuração mais tradicional SO/Hipervisor – MV – Aplicação/Serviço ou como base para Contêineres, PaaS ou *Virtual Appliance* [Dawson 2018].

A este ponto, a importância da virtualização já é justificada através da criação de MVs, contudo, sua aplicabilidade na computação em nuvem vai muito além. A virtualização permite criar infraestruturas lógicas, incluindo topologias de redes e computadores, que funcionam sobre os recursos físicos existentes. Dessa forma, além de criar as MVs, também é possível gerenciar o isolamento e a comunicação entre cada uma das máquinas. Geralmente, esta função é feita por algum software, como hipervisores, por exemplo. No caso do OpenStack, que é o foco deste minicurso, os hipervisores comumente utilizados são o QEMU e o KVM. Por fim, embora a virtualização também possa ser realizada com contêineres, o enfoque deste minicurso está na utilização de máquinas virtuais. Um único servidor pode hospedar diversas instâncias (máquinas virtuais), que são gerenciadas por um hipervisor.

Quando o hipervisor é instalado diretamente sobre o *hardware* (sem a existência de um sistema operacional no computador), tem-se a chamada virtualização Tipo 1, ou *bare metal*. Já na virtualização Tipo 2, o hipervisor executa em um sistema operacional.

A Figura 4.4 ilustra a diferença entre os tipos de virtualização. Assim, em ambos os casos o hipervisor é o principal responsável por todas as operações com as instâncias, como criação, que permite escolher um sistema operacional e criar a instância; edição, que possibilita a alteração de algumas configurações de hardware da instância; salvar estado da máquina (*snapshots*), que armazena informações da instância no momento em que foi solicitado, incluindo uso de memória e armazenamento/disco virtual; e exclusão, que permite a remoção total ou parcial da instância [Chandramouli 2014].

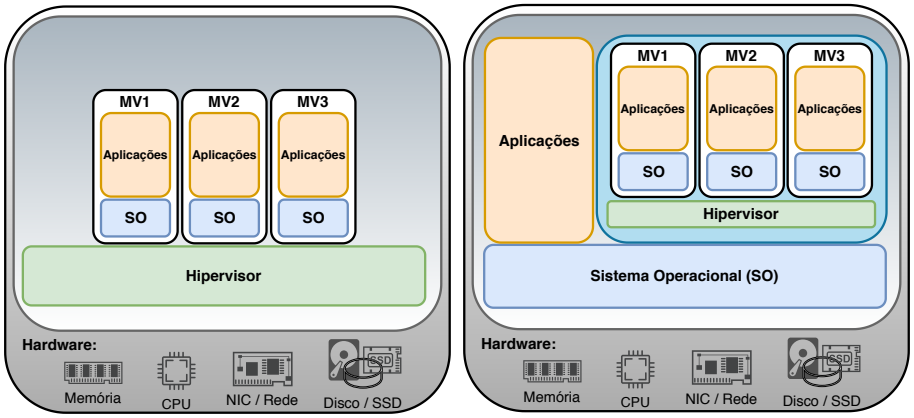

Figura 4.4: Representação dos dois principais tipos de virtualização.

As virtualização possibilitam que diversas MVs sejam hospedadas em um mesmo servidor. Atualmente, hardware para servidores de virtualização possam hospedar dezenas e até mesmo centenas de MVs em um mesmo servidor. Adicionando o fato que as aplicações cada vez tendem a ser mais distribuídas, tem-se a situação na qual a comunicação entre as aplicações/serviços podem ocorrer em MVs que estejam hospedadas em um mesmo servidor e/ou servidores distintos. A Figura 4.5 exemplifica como duas máquinas virtuais podem comunicar-se estando ou não hospedadas no mesmo servidor. Enquanto a comunicação de **MV3** com **MV1** e **MV5** com **MV4** é feita pelo hipervisor, a comunicação de **MV2** com **MV6**, além de passar pelo hipervisor, também passa pela rede física.

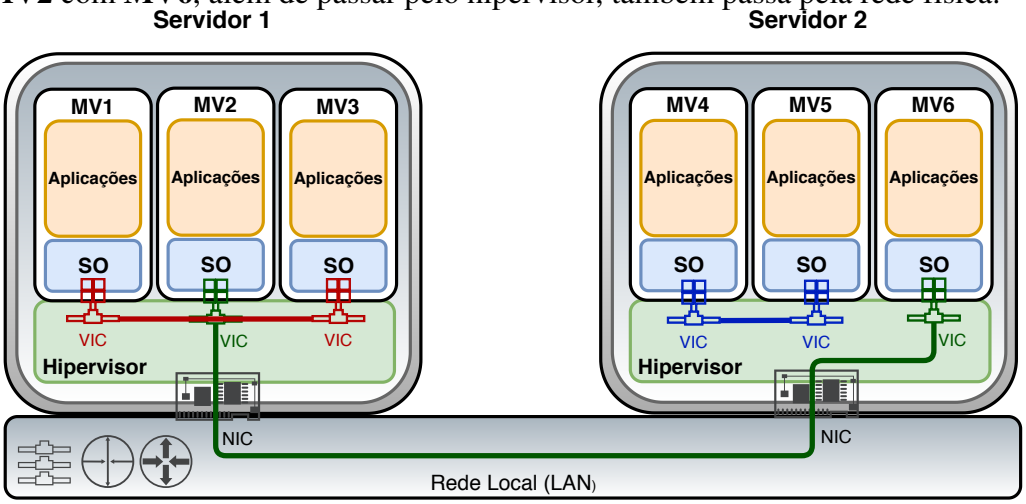

Figura 4.5: Comunicação entre máquinas virtuais estando ou não no mesmo servidor.

Analisando a Figura 4.5, percebe-se que o hipervisor, neste cenário, não fornece apenas a virtualização de computadores (MVs) mas também precisa fornecer soluções

para comunicação entre as MVs, uma vez que nem sempre haverá um recurso de rede físico entre a MV de origem e destino de uma comunicação.

4.1.3. Virtualização de Recursos de Comunicação

A virtualização de recursos de comunicação é amplamente explorada em ambientes de computação em nuvem [Chowdhury and Boutaba 2010]. A Figura 4.5 representa a comunicação entre máquinas virtuais, hospedadas ou não em um mesmo servidor. Quando hospedadas em um mesmo servidor, a comunicação entre as MVs pode ser realizada utilizando roteadores, *switches*, interfaces de rede, *bridges* e endereços *Media Access Control* (MAC) virtualizados, bem como tecnologias tradicionais como *Virtual Local Area Network* (VLAN) e tunelamento *Internet Protocol* (IP). Especificamente, em nuvens OpenStack é recorrente a utilização de *Open vSwitch* [Pfaff et al. 2015] como suporte para a comunicação entre MV.

Em paralelo, o gerenciamento dos recursos de comunicação, virtualizados ou não, podem ser realizados através do uso de *Software Defined Networking* (SDN) [Kreutz et al. 2015, McKeown et al. 2008]. A separação entre os planos de controle e de gerenciamento introduzida por SDN permitiu a composição de infraestruturas virtuais privadas, gerenciadas pelos usuários. Em nuvens OpenStack, o módulo de serviço Neutron faz uso de uma abordagem *Networking as a Service* (NaaS), que através de SDN, fornece serviço de redes aos ambientes computacionais virtualizados. É importante ressaltar que a virtualização de recursos computacionais está presente no oferecimento de serviços aos usuários finais, bem como no gerenciamento da nuvem.

Por outro lado, a virtualização de recursos de comunicação mostra-se fortemente relacionada a implementação do hipervisor e a capacidade deste em usar outras tecnologias/soluções (*e.g.*, *Open vSwitch*, OpenFlow, ...) de modo integrado e facilmente interoperável. Assim, o processo de monitor e analisar tráfego de rede em nuvens computacionais possui uma complexidade ampliada visto que métodos tradicionais (*e.g.*, configurar portas de eco em *switches*, colocar interfaces de rede em modo promíscuo, *etc.*) podem não coletar todo o tráfego.

4.2. Monitoração, análise e caracterização de tráfego

A caracterização de tráfego é uma tarefa usada a fim de entender e resolver problemas que estejam relacionados ao desempenho em redes de computadores [Dainotti et al. 2006]. Neste contexto de nuvens computacionais, a análise e caracterização de tráfego mostra-se como um importante método no levantamento de informações relativas tanto à segurança quanto ao desempenho da nuvem. Através da caracterização de tráfego pode-se, por exemplo, identificar operações em nível de camada de rede bem como o que está trafegando e sua finalidade. Embora esta técnica ainda seja incipiente no contexto da parte de controle em nuvens computacionais, quando se trata de caracterização de tráfego de aplicações tradicionais, como em servidores web, por exemplo, torna-se amplamente empregada [Williamson 2001, Braun and Claffy 1995, Gill et al. 2007].

As nuvens computacionais em geral têm dois locais nos quais a monitoração e análise do tráfego podem ser empregadas. Em primeiro lugar, a rede dos usuários pode ser monitorada a fim de aplicar controles de privacidade e uso da nuvem. Além disso,

também é possível monitorar o tráfego administrativo da nuvem. Desse modo, alguns recursos, tanto computacionais quanto de redes e comunicação, podem ser melhor dimensionados de acordo com o tráfego administrativo analisado, *e.g.*, aumentar a largura de banda quando necessário.

De maneira geral, o estudo do tráfego é separado em duas etapas: medição e análise [Dainotti et al. 2006]. Na primeira, tem-se a coleta de dados que trafegam na rede, enquanto na segunda, é feita uma análise do tráfego para identificar características relevantes ao problema. No contexto deste minicurso, estaremos coletando e analisando o tráfego gerado na rede administrativa do OpenStack (Seção 4.3) durante o ciclo de vida induzido das máquinas virtuais. O objetivo aqui é identificar, a partir do tráfego gerado, quais serviços do OpenStack estão em execução no momento, bem como, qual o volume de tráfego gerado durante o processo. Assim, estas informações podem ser utilizadas no gerenciamento proativo da rede, como por exemplo, criar uma rede isolada para tráfego das imagens das instâncias.

Durante a etapa de medição de tráfego, podem ser utilizadas diversas ferramentas para capturar os dados que trafegam pela rede (*e.g.*, TCPdump e SNORT), contudo, durante os experimentos aqui descritos, utilizou-se uma ferramenta autoral baseada em TCPdump, que coleta e armazena os pacotes. Além disso, dependendo de como é implementada, a medição pode ser feita de forma passiva ou ativa [Vilela 2006, Williamson 2001]. O método passivo é aplicado em casos onde não se deseja intrusão na rede. Não são feitas injeções de tráfego ou quaisquer outras alterações na rede. Já no método ativo, o tráfego na rede pode ser influenciado pela ferramenta de medição.

Em relação a etapa de análise, aplicam-se técnicas de análise de tráfego, como a modelagem estatística simples, que permite analisar atributos genéricos do tráfego e correlacioná-los com as aplicações correspondentes que estão em execução [Nguyen and Armitage 2008]. Neste sentido, são utilizados atributos como as portas de origem e destino dos pacotes, que indicam o serviço correspondente a estes pacotes. Além disso, é importante notar que o processo de caracterização de tráfego é consideravelmente variável de acordo com o ambiente (configuração de rede). Dessa forma, as Seções 4.3 e 4.4 trazem informações sobre o funcionamento do OpenStack, bem como, o ambiente no qual foram realizados os experimentos.

4.3. OpenStack

O OpenStack é um conjunto de *software* e ferramentas de código aberto para criar e gerenciar nuvens computacionais públicas e privadas do tipo IaaS. De acordo com o OpenStack Project [OpenStack 2019b], trata-se de um sistema operacional para nuvens, que controla um amplo grupo de recursos de computação, armazenamento e rede por todo o *data center*. Atualmente, o OpenStack está na sua 19ª versão (denominada Stein).

O OpenStack tornou-se uma solução de nuvem amplamente utilizada no mundo devido a quantidade de empresas envolvidas no seu desenvolvimento (`https://www.openstack.org/analytics`). Tal fato fez com que as principais soluções de MVs (*e.g.*, KVM, QEMU, VMware ESX/ESXi, Citrix Xen, Microsoft Hyper-V, UML, Virtuozzo, IBM zVM, ...) fossem incorporadas aos seus serviços de computação. Da mesma forma foram incorporados serviços de virtualização de comunicação (*e.g.*, SDN com

OpenFlow, ...) e armazenamento (*e.g.*, Gluster, CEPH, ...).

4.3.1. Serviços OpenStack

Os serviços do OpenStack, usados para o provisionamento de nuvens computacionais, são separados por módulos. Além disso, é possível que módulos opcionais sejam acoplados a uma instalação. A Figura 4.6 mostra todos os principais módulo de serviço do OpenStack, bem como serviços e recursos externos que podem ser incorporados a solução de nuvem

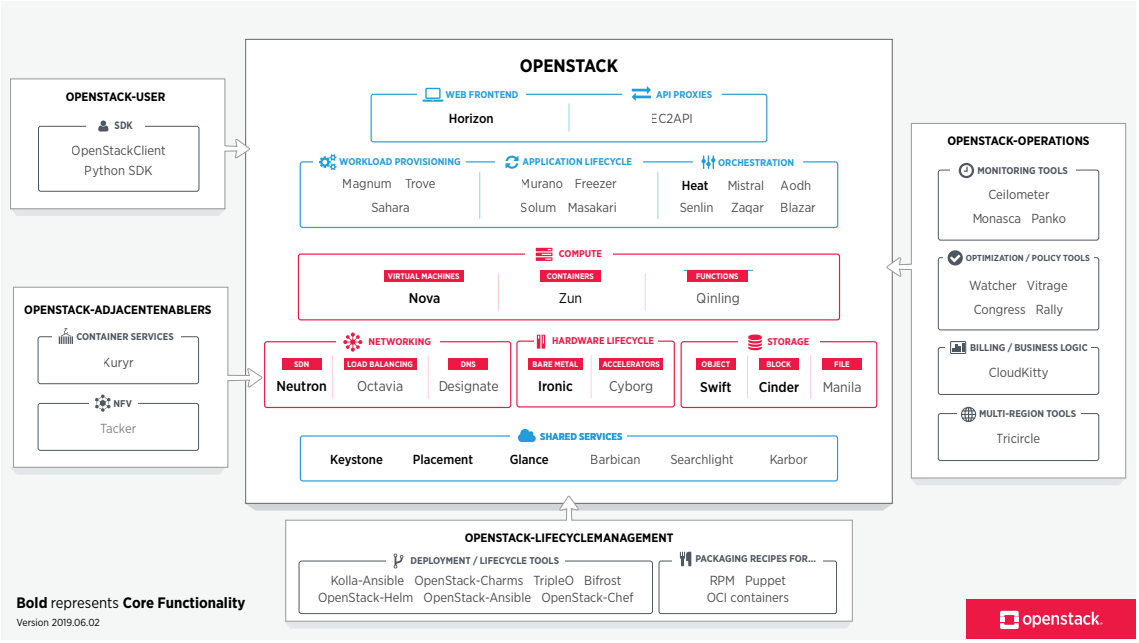

Figura 4.6: Visão geral dos módulo de serviço disponíveis no OpenStack [OpenStack 2019a]

Módulos mais comuns do OpenStack presentes nas instalações tradicionais:

- **Horizom**: Disponibiliza um serviço de *dashboard web* para uso e administração da nuvem;
- **Keystone**: Disponibiliza um serviço de autenticação e autorização para outros serviços do OpenStack;
- **Nova**: Responsável pela distribuição e gerenciamento das instâncias. Realiza tarefas como iniciação, escalonamento e desalocação de MVs;
- **Neutron**: Fornece conectividade de rede entre os outros serviços, bem como disponibiliza uma *Application Programming Interface* (API) para que os consumidores configurem suas redes;
- **Glance**: Atua no processo de armazenamento e recuperação das imagens utilizadas nas MVs;
- **Swift**: Responsável pelo armazenamento e recuperação de objetos não estruturados. Usa técnicas de replicação de dados para tolerância de falhas; e

- **Cinder**: Provê armazenamento persistente em bloco para instâncias em execução.

A Figura 4.7 exemplifica como os principais módulos do OpenStack interagem entre si e com uma MV. Pode-se concluir que, de acordo com a definição da instalação da nuvem, alguns deste módulos podem estar em um mesmo servidor ou em servidores distintos.

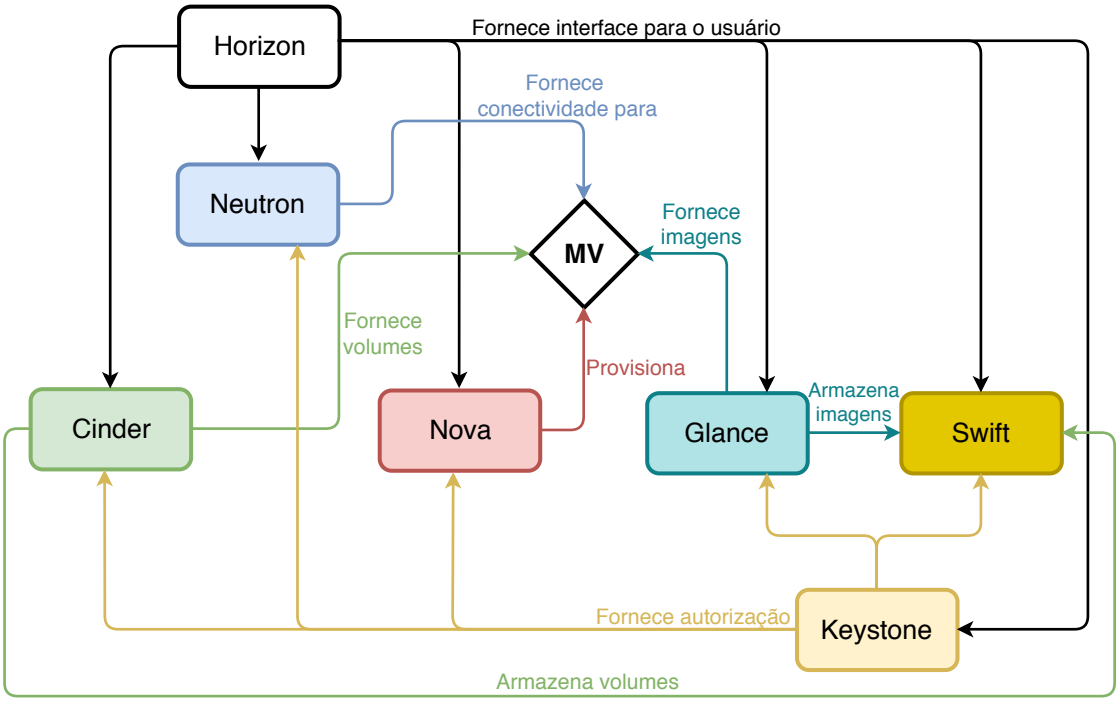

Figura 4.7: Principais operações dos módulo sobre uma MV.

Em implantações do OpenStack em cenário de produção, os módulos são distribuídos entre os servidores da nuvem de modo que existam ganhos com desempenho e segurança da nuvem. Ou seja, serviços computacionalmente intensivos são posicionados em servidores distintos buscando diminuir a interferência no desempenho final do OpenStack. Já em implantações menores, como em uma prova de conceito, os módulos acabam centralizados em um mesmo servidor e os demais servidores são destinados a serviço de computação (*e.g.*, Nova).

4.3.2. Configuração de Rede

Para realizar a comunicação entre os servidores que hospedam diferentes serviços são utilizadas múltiplas redes físicas e virtuais. Neste sentido, existem três domínios criados de acordo com políticas de segurança distintas [OpenStack 2019b]: Domínio Público, Domínio de Controle e Domínio de Convidados. A Figura 4.8 exemplifica a infraestrutura de comunicação recomendada como padrão pelo OpenStack. Podem ser utilizadas técnicas de virtualização de redes, como VLAN, para separação dos domínios de rede.

Domínio Público: são englobadas as redes Externa e de API, que são responsáveis pela visibilidade da API da nuvem criada na Internet e do acesso à Internet pelas MVs. Todos

os endereços IPs das redes devem ser visíveis a partir da Internet para seu pleno funcionamento [OpenStack 2019b].

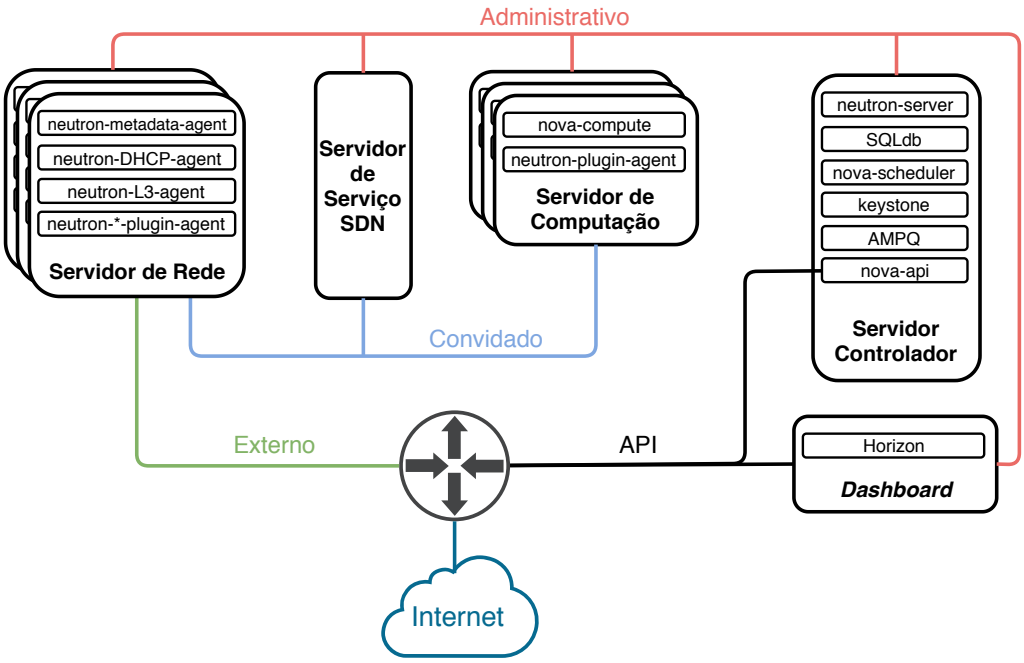

Figura 4.8: Configuração de rede recomendada pelo OpenStack [OpenStack 2019a].

Domínio de Convidados é formado pela rede de comunicação entre as MVs. Os consumidores do serviço de oferta de MVs são referenciados como convidados por não possuírem vínculo direto com a administração da nuvem, logo, não há garantias sobre quem estes convidados são e quais os seus objetivos. Além disso, através do serviço Neutron, são utilizadas tecnologias de virtualização de rede tanto para isolar o tráfego entre as diversas MVs hospedadas em um servidor, quanto para criar redes virtuais entre elas e tornar possível a comunicação quando assim configurado pelo consumidor [OpenStack 2019b].

Domínio de Controle: é formado principalmente pela rede responsável pelo tráfego de controle do OpenStack, que corresponde ao tráfego gerado pela comunicação entre os seus serviços. Esta é a rede mais interna da nuvem, sendo que somente os administradores devem acessá-la. Portanto, esta rede dispensa o uso de técnicas de virtualização da rede. Além disso, tanto os servidores de controle quanto os de computação devem ter uma interface física de rede que conecta-se com a rede de controle. O uso de múltiplas interfaces de rede permite a conectividade à rede de controle e as outras redes conforme o caso, como a rede de convidados no caso dos servidores de computação. Logo, em casos normais, uma MV em execução nunca poderá acessar a rede de controle. [Krutz and Vines 2010].

4.3.3. Fluxo de Dados para Criação de Máquinas Virtuais

O OpenStack permite que sejam feitas diversas operações com as instâncias de MVs, como criar, excluir e armazenar. Toda vez que o usuário realizar uma operação com a instância, o estado dela é alterado de acordo com a operação realizada. Por exemplo, ao remover uma instância, será aplicado o estado DELETED quando a operação for completada. Neste sentido, a documentação fornece uma lista com todas as possíveis operações e estados associadas às instâncias, contudo, neste minicurso serão utilizadas as operações principais para manipulação de uma MV: *CREATE()*, *SUSPEND()*, *RESUME()*, *STOP()*

e *SHELVE()* [OpenStack 2019b].

A Figura 4.9 mostra os possíveis estados que uma instância de máquina virtual pode assumir de acordo com as operações realizadas. Com exceção de BUILDING, que representa a construção inicial da instância, todos os demais estados apresentados nesta figura são chamados de "VM_STATE", pois, indicam o estado estável e atual da instância. Dessa forma, o estado ACTIVE, que indica que a instância está em execução com a sua respectiva imagem, será atingido após as operações *CREATE()* e *RESUME()* serem executadas. Já o estado SUSPENDED, indica a suspensão da instância em nível de hipervisor, ou seja, o estado atual da máquina é mantido, mas é salvo na unidade de armazenamento, não em memória, como é o caso do estado PAUSED. Em relação ao STOPPED, a máquina encontra-se parada e, todos os recursos associados a ela são desalocados. Isso indica que a instância não pode ser restaurada no seu estado anterior. Quanto ao SHELVED, este estado indica que a MV encontra-se desligada e armazenada para uso posterior. Assim, todos os dados e recursos relacionados a ela são mantidos, como volumes adicionados, por exemplo, contudo, as informações em memória são descartadas [OpenStack 2019a].

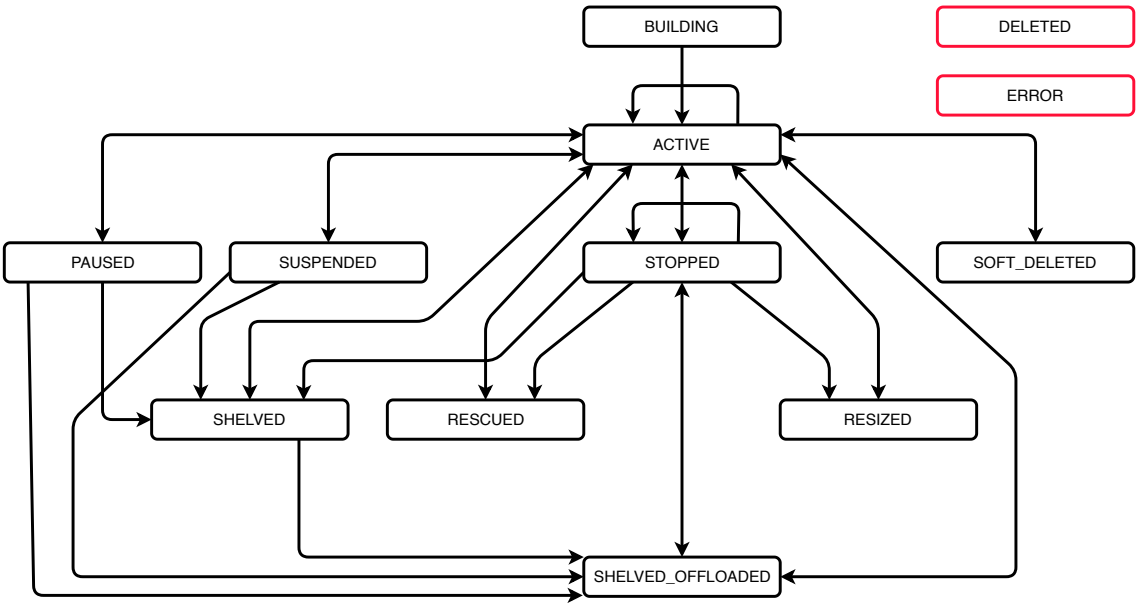

Figura 4.9: Transições de estados para o gerenciamento de uma MV em OpenStack [OpenStack 2019a]

O OpenStack também fornece o estado das tarefas das instâncias, chamado de "TASK_STATE", que não deve ser confundido com o "VM_STATE". O estado da tarefa sempre indica uma ação em execução no momento (transição de estado), enquanto o "VM_STATE"indica uma ação que já foi concluída (estado estável atual da MV). Por exemplo: a operação *SUSPEND()* aplica um "TASK_STATE" SUSPENDING e, ao finalizar, um "VM_STATE"SUSPENDED. Além disso, quando não há nenhuma tarefa sendo executada na instância, aplica-se o "TASK_STATE"*None* a ela. Durante o minicurso será dado maior enfoque ao "VM_STATE", ainda assim, é importante entender as diferenças entre o estado da tarefa e da MV.

Em termos de comunicação e volume de tráfego gerado na rede, as transições

de estado mais custosas são aquelas que requisitam ou persistem dados relacionados às instâncias de MVs. Dessa forma, de acordo com a configuração mostrada na Figura 4.8, o servidor de computação, no qual a instância será hospedada, encontra-se separado dos demais, o que significa que a comunicação pela rede é necessária. Neste sentido, como neste caso não existe uma rede dedicada para armazenamento (*Storage*), além de toda comunicação inter-serviço, a rede administrativa será utilizada como meio de transmissão das imagens das instâncias, por exemplo. Então, operações como *CREATE()* e *SHELVE()* têm maior custo, no sentido de haver maior comunicação entre os serviços e gerar mais tráfego na rede administrativa.

Por fim, para entender melhor o fluxo de dados para criação de máquinas virtuais, alguns processos internos do OpenStack devem ser conhecidos. Inicialmente, é necessário uma imagem do sistema operacional que será executado na instância. Então, conforme a instalação do OpenStack, essa imagem será armazenada no Swift ou no Glance, que serão encarregados por distribuí-la ao servidor de computação quando assim requisitado. Além disso, outro processo importante é o de autenticação e autorização na nuvem, que é realizado pelo Keystone, como descrito na Subseção 4.3.1. Sem autorização prévia, a criação da instância não é permitida. Também é válido notar que, durante o provisionamento da instância, existem trocas de mensagens entre os serviços do OpenStack, como chamadas API feitas para requisitar informações de outros módulos, por exemplo.

4.3.4. Monitoração em OpenStack com Foco na Rede de Controle

A maior parte das pesquisas preocupa-se em analisar as redes da nuvem do ponto de vista dos usuários, deixando de fora toda a parte de operações internas e comportamento do provedor da nuvem [Aishwarya. K and Sankar 2015, Chen and Zhao 2012, Shete and Dongre 2017]. O dimensionamento e controle de tráfego na rede administrativa de uma nuvem OpenStack depende da estratificação de uso e, consequente impacto de operações de controle oriundas de demandas de ações dos usuários. A ausência de monitoração ou caracterização de tráfego na rede administrativa de uma nuvem OpenStack pode causar colapsos e gargalos na rede degradando a qualidade ou mesmo a indisponibilidade do serviço. Entre as operações mais comuns de um usuário de nuvem estão as de gerência de MV, foco de análise de trabalho. Portanto, o tráfego coletado é oriundo das operações criação, remoção, suspensão, etc em MVs. Para isso, seguiu-se um ciclo de vida induzido das instâncias, que as faz transitar entre os estados BUILDING, ACTIVE, SUSPENDED, STOPPED e SHELVED. A Figura 4.10 ilustra este ciclo, bem como mostra as operações realizadas para cada transição de estado. Por fim, será possível dizer quanto tempo cada operação levou para executar, qual o volume de tráfego gerado por ela e quais chamadas de API foram feitas durante o processo.

4.4. Ambiente de testes

Para que o objetivo final de análise e caracterização do tráfego seja atendido, os testes consistem em submeter instâncias a um ciclo de vida induzido, baseado nas operações mais comuns em máquinas virtuais, que consiste em: criação; suspensão; reativação (*RESUME()*); parada; e armazenamento (*SHELVE()*). Dessa forma, cada operação resultará em um volume de tráfego gerado no domínio de controle do OpenStack. Assim, através de uma ferramenta autoral baseada em TCPdump, pode-se coletar e posteriormente

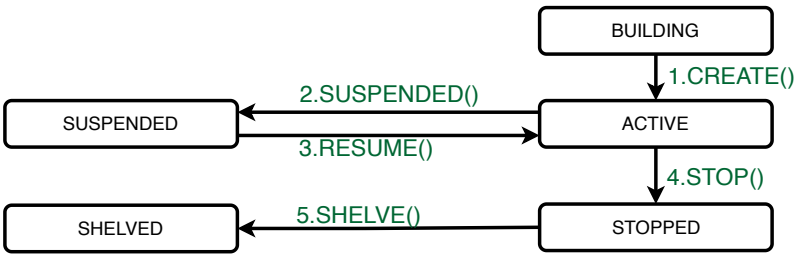

Figura 4.10: Ciclo de vida induzido das máquinas virtuais.

analisar esse tráfego, de modo a identificar o que está trafegando na rede e sua finalidade.

Todos os experimentos são executados em uma instalação do OpenStack Queens, cuja configuração segue o modelo de prova de conceito com dois servidores (Figura 4.11). A infraestrutura da nuvem é provida pelo CloudLab (`https://cloudlab.us/`), que fornece dois servidores, um para computação e outro para controle da nuvem. Cada um dos servidores conta com 256GB de memória e dois processadores que totalizam 16 núcleos [CloudLab 2019].

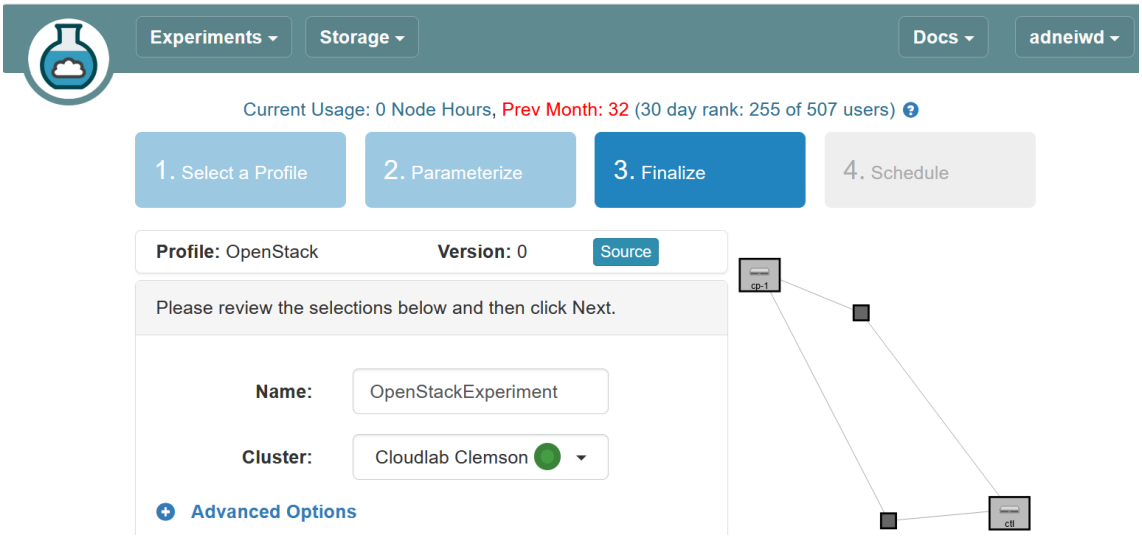

Figura 4.11: Ambiente do CloudLab para criação de um experimento com o OpenStack [CloudLab 2019].

4.5. Estudo de caso com o OpenStack

Inicialmente, os experimentos aqui descritos foram aplicados em uma instalação do OpenStack Queens em um ambiente provido pelo CloudLab. Cada experimento consiste em: (1) identificar a rede administrativa; (2) executar o ciclo de vida induzido das MVs; (3) coletar o tráfego administrativo durante toda a execução; (4) gerar um relatório de transição de estados das máquinas; e (5) analisar o tráfego referente a cada operação.

Na Etapa 1, identifica-se a rede através de testes manuais. Cria-se uma instância na nuvem e verifica-se o volume de tráfego gerado em cada uma das VLANs. A imagem da instância irá trafegar pela rede administrativa. Dessa forma, quando a instância é

criada, a VLAN administrativa terá volume de tráfego próximo ao tamanho da imagem da instância. Como neste ambiente existem apenas duas VLANs, a outra será a pública.

A Figura 4.12 mostra a configuração do ambiente. As Etapas 2 e 3 são feitas por uma ferramenta autoral implementada em Python 3.6, na qual cada operação do ciclo de vida induzido das instâncias é feita através das APIs para Python do OpenStack. Em relação à coleta de tráfego, a ferramenta baseia-se na utilização de TCPdump, e todo tráfego referente a interface escolhida é coletado durante a execução do experimento. A técnica de medição aqui adotada é passiva, visto que não há intrusão na rede (Seção 4.2). Contudo, como o tráfego presente na rede é produto das operações executadas pelo algoritmo, o método de medição também pode ser interpretado como híbrido. A Etapa 4 utiliza o banco de dados do OpenStack, mais especificamente do serviço Nova, para identificar com precisão as transições de estado de cada máquina virtual. Então, fica possível classificar o tráfego coletado referente a uma determinada operação. Por fim, na Etapa 5, o tráfego é analisado de modo a identificar os serviços em execução no momento, como conceituado na Seção 4.2 e detalhado na Subseção 4.3.4.

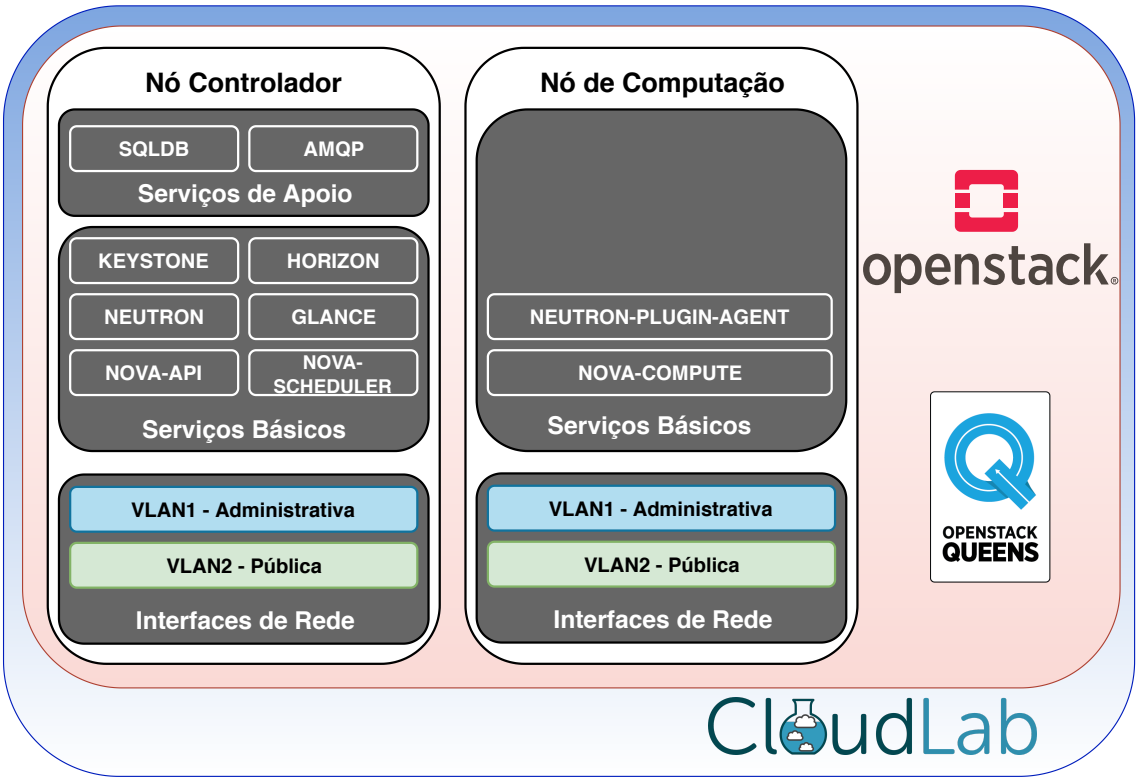

Figura 4.12: Configuração utilizada durante os experimentos, redes Pública e Administrativa são isoladas em VLAN.

4.5.1. Resultados

O processo de experimentação explicado foi aplicado utilizando uma imagem do sistema operacional GNU/Linux CentOS 7, com formato "QCOW2"e tamanho de 1.3GB. O gráfico da Figura 4.13 mostra uma linha do tempo da execução do experimento. A criação da instância de MV é iniciada no segundo 1 e finalizada no segundo 35; a operação

SUSPEND() ocorre entre os segundos 36 e 42; já a operação *RESUME()* se dá entre os segundos 43 e 44; enquanto a *STOP()* vai do segundo 45 até 46; e, a operação *SHELVE()* é executada entre os segundos 47 e 81.

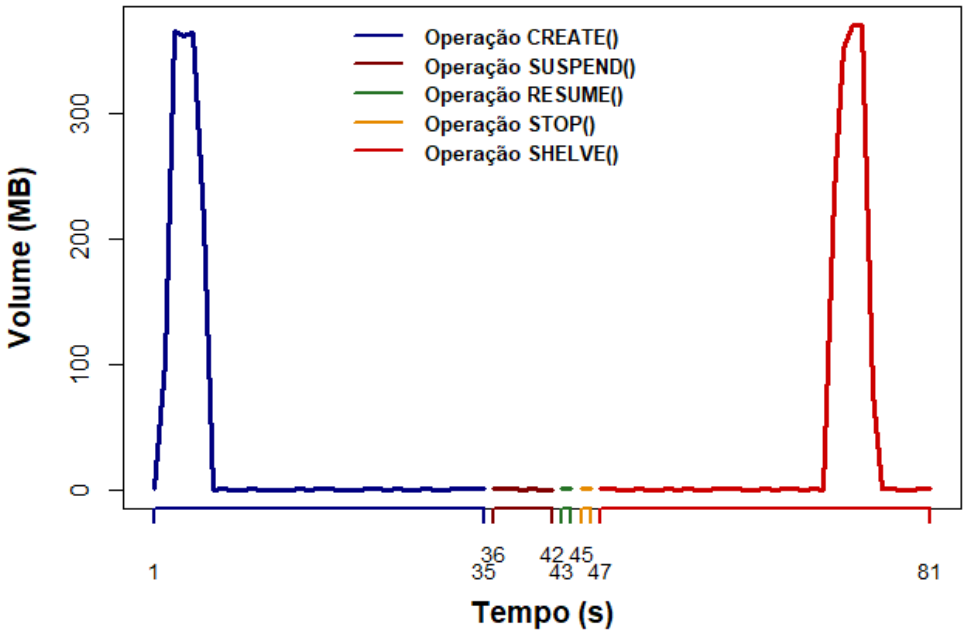

Figura 4.13: Tráfego coletado por segundo durante toda a execução do experimento.

Dessa forma, na Figura 4.13, nota-se que as operações *CREATE()* e *SHELVE()*, além de terem maior período de duração, também são as que produzem maior volume total de tráfego. Além disso, são observados dois picos, um entre 2 e 5 segundos, na operação *CREATE()*, e outro na operação *SHELVE()*, entre 71 e 75 segundos. O somatório de tráfego no intervalo de ambos os picos é de cerca de 1.4GB, o que indica que, muito provavelmente, durante esses intervalos ocorreram as transferências da imagem utilizada na máquina virtual (GNU/Linux CentOS 7) e dos demais recursos associados.

Relacionando os dados do gráfico da Figura 4.13 com a Tabela 4.1, é possível notar que o serviço Glance, que é responsável pelo gerenciamento das imagens das instâncias é o maior responsável pelo volume de tráfego gerado, o que reforça a explicação aos intervalos de pico no gráfico. Na sequência, tem-se a categoria de tráfego "ETC", assim chamada pois, engloba o tráfego de rede referente a serviços diversos, que em sua maioria, é composto pelo RabbitMQ, um software utilizado para troca de mensagens entre os serviços do OpenStack.

Tabela 4.1: Média de tráfego por serviço coletado na interface da rede administrativa do OpenStack. Valores em MB.

	Glance	**Nova**	**Keystone**	**Neutron**	**ETC**	**Total**
CREATE	1398,116	0,003749	0,082098	0,032799	21,04559	1419,28
SUSPEND	0,000208	0,00117	0,011826	0,000104	3,851485	3,864793
RESUME	0	0,001169	0	0,007377	1,453582	1,462128
STOP	0	0,00117	0,011781	0	1,274953	1,287904
SHELVE	1400,127	0,001377	0,023681	0,014858	19,64397	1419,811

Ao analisar o tráfego coletado durante o experimento, também notou-se a presença de diversas chamadas de API do OpenStack. Como mostra a Tabela 4.2, durante a operação de criação da instância, foram realizadas 3 chamadas ao serviço Nova, 13 ao Neutron, 3 ao Keystone e 1 ao Glance. Em relação às operações SUSPEND e STOP, ambas tiveram apenas uma chamada ao serviço Keystone e uma chamada ao Nova. Já a operação RESUME teve apenas uma chamada de API, que foi feita ao Nova. Enquanto a operação SHELVE foi a que teve maior número de chamadas API, com 10 chamadas ao Glance, 3 ao Keystone, 10 ao Neutron e 1 ao Nova.

Tabela 4.2: Número de chamadas API por serviço de acordo com a operação feita na instância.

	Glance	**Nova**	**Keystone**	**Neutron**
CREATE	1	3	3	13
SUSPEND	0	1	1	0
RESUME	0	1	0	0
STOP	0	1	1	0
SHELVE	10	1	3	10

Por fim, de acordo com o módulo de relatório das MVs, todas as operações foram executadas com êxito e, as operações CREATE e SHELVE foram as duas mais demoradas, levando em torno de 34 segundos de execução cada uma. A operação SUSPEND levou em torno de 6 segundos, enquanto a RESUME e STOP levaram apenas 1 segundo cada uma. A Tabela 4.3 mostra os tempos de execução para cada operação.

Tabela 4.3: Tempo de execução de cada operação em segundos.

Tempo de execução em segundos				
CREATE	**SUSPEND**	**RESUME**	**STOP**	**SHELVE**
34	6	1	1	34

4.6. Considerações

As pesquisas mais recentes mostram um considerável interesse em analisar as redes da nuvem que são visíveis aos usuários. Dessa forma, as operações que acontecem no provedor da nuvem acabam não recebendo a devida atenção. Neste sentido, a falta de informações relacionadas ao desempenho do provedor motiva a análise e caracterização do tráfego gerado pelas operações realizadas sobre instâncias de máquinas virtuais (MVs) na nuvem. Inicialmente, de acordo com os resultados mostrados da Subseção 4.5.1, fica visível que as operações de criação e armazenamento das instâncias fazem maior uso da rede. Neste sentido, para que não existam problemas de *Quality of Service* (QoS) ou SLAs, os provedores devem estar preparados para atender a demanda de criação de múltiplas instâncias ao mesmo tempo. Assim, por meio da análise do tráfego gerado por esta operação, é possível planejar a distribuição e configuração dos recursos de rede da nuvem, de modo que o desempenho não seja afetado pela alta demanda de serviço.

Além disso, os resultados também apontam que a maior parte do volume de tráfego gerado é causado pela transferência da imagem do sistema operacional da instância.

Então, caso necessário, a configuração de rede da nuvem pode ser revisada, com o objetivo de diminuir o tráfego de imagens na rede de controle da nuvem. Os resultados também mostram uma significante quantia de tráfego caracterizado como "ETC", pertencente a serviços diversos. Dessa forma, é necessário estudar quais são os serviços diversos em questão e qual o seu impacto na rede administrativa.

Agradecimentos

Os autores agradecem o apoio do Laboratório de Processamento Paralelo Distribuído (LabP2D) no Centro de Ciências Tecnológicas (CCT) da Universidade do Estado de Santa Catarina (UDESC).

Os autores agradecem o apoio da Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina (FAPESC).

Referências

[Aishwarya. K and Sankar 2015] Aishwarya. K and Sankar, S. (2015). Traffic analysis using hadoop cloud. In *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, pages 1–6.

[Bohn et al. 2011] Bohn, R. B., Messina, J., ai Liu, F., Tong, J., and Mao, J. (2011). Nist cloud computing reference architecture. *2011 IEEE World Congress on Services*, pages 594–596.

[Braun and Claffy 1995] Braun, H.-W. and Claffy, K. C. (1995). Web traffic characterization: an assessment of the impact of caching documents from ncsa's web server. *Computer Networks and ISDN Systems*, 28(1):37 – 51. Selected Papers from the Second World-Wide Web Conference.

[Chandramouli 2014] Chandramouli, R. (2014). *Security recommendations for hypervisor deployment*. US Department of Commerce, National Institute of Standards and Technology.

[Chen and Zhao 2012] Chen, D. and Zhao, H. (2012). Data security and privacy protection issues in cloud computing. In *2012 International Conference on Computer Science and Electronics Engineering*, volume 1, pages 647–651.

[Chowdhury and Boutaba 2010] Chowdhury, N. M. K. and Boutaba, R. (2010). A survey of network virtualization. *Computer Networks*, 54(5):862–876.

[CloudLab 2019] CloudLab (2019).

[Dainotti et al. 2006] Dainotti, A., Pescape, A., and Ventre, G. (2006). A packet-level characterization of network traffic. In *2006 11th International Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Networks*, pages 38–45.

[Dawson 2018] Dawson, M. (2018). Red Hat Global Customer Tech Outlook 2019: Automation, cloud, & security lead funding priorities.

[Gill et al. 2007] Gill, P., Arlitt, M., Li, Z., and Mahanti, A. (2007). Youtube traffic characterization: A view from the edge. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 15–28, New York, NY, USA. ACM.

[Jadeja and Modi 2012] Jadeja, Y. and Modi, K. (2012). Cloud computing - concepts, architecture and challenges. In *2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*, pages 877–880.

[Kreutz et al. 2015] Kreutz, D., Ramos, F. M. V., Veríssimo, P. E., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1):14–76.

[Krutz and Vines 2010] Krutz, R. L. and Vines, R. D. (2010). *Cloud security: A comprehensive guide to secure cloud computing*. Wiley Publishing.

[McKeown et al. 2008] McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.

[Mell and Grance 2011] Mell, P. M. and Grance, T. (2011). Sp 800-145. the nist definition of cloud computing.

[Nguyen and Armitage 2008] Nguyen, T. T. T. and Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys Tutorials*, 10(4):56–76.

[OpenStack 2019a] OpenStack (2019a). Openstack documentation.

[OpenStack 2019b] OpenStack (2019b). What is openstack?

[Panizzon et al. 2019] Panizzon, G., Battisti, J. H. F., Koslovski, G. P., Pillon, M. A., and Miers, C. C. (2019). A Taxonomy of container security on computational clouds: concerns and solutions. *Revista de Informática Teórica e Aplicada*, 26(1):47–59.

[Pfaff et al. 2015] Pfaff, B., Pettit, J., Koponen, T., Jackson, E. J., Zhou, A., Rajahalme, J., Gross, J., Wang, A., Stringer, J., Shelar, P., Amidon, K., and Casado, M. (2015). The design and implementation of open vswitch. In *Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation*, NSDI'15, pages 117–130, Berkeley, CA, USA. USENIX Association.

[project 2019] project, A. C. (2019). Apache cloudstack open source cloud computing.

[Scarfone et al. 2011] Scarfone, K., Souppaya, M., and Hoffman, P. (2011). Guide to security for full virtualization technologies. *NIST Special Publication*, 800:125.

[Shete and Dongre 2017] Shete, S. and Dongre, N. (2017). Analysis amp; auditing of network traffic in cloud environment. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 97–100.

[Vilela 2006] Vilela, G. S. (2006). Caracterização de tráfego utilizando classificação de fluxos de comunicação. Mestre em ciências em engenharia de sistemas e computação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil.

[Williamson 2001] Williamson, C. (2001). Internet traffic measurement. *IEEE Internet Computing*, 5(6):70–74.