Use Software Defined Networking to optimize your IaaS

Manage networking through software abstraction layers to enhance your cloud infrastructure

Judith M. Myerson June 24, 2014

Explore Software Defined Networking (SDN) — network management via software abstraction layers — as a method to enhance and optimize your Infrastructure as a Service in the areas of interoperability, user and provider expectation management, developer and administrator requirements, and effective risk mitigation.

Software-defined networking (SDN) is a networking approach that allows administrators to manage network services by abstracting lower-level functionality. SDN decouples the **control plane**— which determines where traffic is sent — from the **data plane**— which forwards the traffic to the chosen destination.

In this article, I explain the combination of SDN with cloud infrastructure services in order to optimize the IaaS; I concentrate on these areas:

- Ensuring IaaS interoperability
- Fully utilizing the IaaS cloud service model
- Meeting user, developer, provider, and maintainer expectations with OpenStack Foundation and OpenDayLight projects
- Delivering cost-effective mitigation of risks to IaaS optimization

IaaS interoperability

A network administrator can accomplish the goal of IaaS interoperability by using the SDN architecture with **Network Functions Virtualization** (NFV). The NFV architecture concept calls for using virtualization technologies to virtualize entire classes of network node functions into building blocks that can then be interconnected to create communication services. One of these virtualized network functions may be composed of one or several virtual machines that might be running different software and processes atop a cloud infrastructure.

The SDN architecture lets the administrator have a global view of how the traffic behaves from one network device to another in an IaaS from a central console; when necessary, it also details how traffic should be optimized before moving it to a compatible IaaS.

The core of the SDN architecture is the **controller,** an open source application that separates control functions from the data functions of the proprietary network devices. The administrator can use the controller to tell network devices where to redirect or resend the packets if one network device begins to experience heavy traffic. The administrator can add to the IaaS VMs when needed: for example, to move massive amounts of data from poorly performing network devices to more healthy ones.

NFV decouples the network functions — such as network address translation (NAT), firewalling, intrusion detection, domain name service (DNS), and caching — from proprietary hardware appliances so that the functions can run the controls in software. It consolidates the networking components needed to support a fully virtualized networking infrastructure.

The SDN and NFV controller must be properly configured in order for the administrator to optimize or make other changes to the network traffic flows. Improper configuration can lead to IaaS outages or attacks. Proper configurations can be accomplished through such open source projects as OpenStack and OpenDayLight.

To find out how many networking pros prefer open source for their SDN solutions from commercial suppliers, OpenDayLight conducted a survey ... The report surveyed 600 IT decision makers and technologists in medium to large organizations within enterprise (300) and service provider (300) organizations in North America. The survey shows that 95% of networking pros want open source in their SDN solutions, but 76% of them prefer open source from commercial suppliers. ("SDN, NFV Open Source Report: The Operator's View.")

By choosing open source from commercial suppliers, SDN and NFV controller can be used to optimize the IaaS.

IaaS cloud service model

In order to better understand how IaaS optimization with SDN would work, you need to understand how the IaaS cloud service model major players are compared with one another in controlling an IaaS cloud. The important participants include:

- IaaS cloud service users
- PaaS/IaaS cloud service developers
- IaaS cloud service providers
- SDN administrators

IaaS cloud service users

Let's take a look at the extent of control IaaS cloud service users have. This group includes IaaS users and SaaS users.

The **IaaS user** (usually a network or infrastructure specialist):

- Controls the operating systems, network equipment, and deployed applications at the virtual machine level.
- The infrastructure specialist can scale up or down the virtual services or blocks of storage area.
- He or she may also be the SDN administrator who uses the controller to optimize the traffic flow from one device to another.

By contrast, the only control the **SaaS user** (whether he or she is a private individual, business, or government agency) has is to access the SaaS application; the SDN administrator's control of traffic flow is transparent to him or her. The SaaS user can also be the SaaS provider who takes into account on meeting the user's expectations on access control, system, response time (throughput), and inquiry responses.

PaaS/IaaS cloud service developers

How much control do the PaaS and IaaS cloud service developers have?

The **PaaS developer** controls and protects all the applications found in a full business lifecycle. The developer builds, deploys, and runs, say, a custom warehouse management application. As part of the business lifecycle, the developer uses spreadsheets, word processors, billing, payroll processing, and invoicing. The PaaS developer needs to make sure the application would work well in the networks over which the SDN administrator has control.

The **IaaS developer** controls the lifecycle development of an IaaS public or private cloud that can interoperate with another IaaS cloud hosted by a different service provider. The IaaS developer ensures that the IaaS is based on OpenStack. The IaaS developer works with the PaaS developers on running SaaS applications on the IaaS or PaaS in the testing environment.

IaaS cloud service providers

At a minimum, the IaaS provider controls the infrastructure of traditional computing resources underlying virtual machines. The provider sets the user, resource, and data requests threshold levels and might allow negotiation on changing the threshold levels with the PaaS developers. The provider might also be the SDN administrator who controls the traffic between network devices.

SDN administrators

In an IaaS cloud, the administrator has control over SDN functions decoupled from the data plane and NFV functions separated from the proprietary hardware appliances. The administrator will need to collaborate with IaaS cloud service model players on optimizing the open-source IaaS.

Meeting expectations: OpenStack Foundation

Prospective and current cloud service customers expect an open cloud service standard to allow one IaaS to interoperate with another IaaS hosted by another provider. OpenStack took proactive steps to meet their expectations: The OpenStack Foundation Project allows developers and cloud

computing technologists to collaborate on producing the open source cloud computing platform for IaaS public and private clouds.

Let's take a little closer look at what OpenStack Foundation is about and what the foundation has been doing with OpenStack to standardize the IaaS. Then I'll discuss modular architecture components, shared services, and Neuron and SDN.

The Foundation's work

OpenStack is an IaaS cloud operating system that controls large pools of compute, storage, and networking resources and shared services throughout a data center. All are managed through a dashboard that gives administrators control while empowering their users and developers to provision resources through a web interface.

OpenStack Foundation oversees the OpenStack project that integrates code from NASA's Nebula platform with Rackspace's platform. Code changes to this project are contributed by members of the OpenStack Foundation that was spun off in 2011 from Rackspace. Developers and cloud computing technologists collaborate globally to produce open source cloud computing platform for public and private clouds.

In April 2012, IBM and Red Hat agreed to join the foundation as platinum members, meaning they will contribute US$500,000 a year for the next three years. Other companies planning to sign on as platinum members include AT&T, Canonical, HP, Nebula, Rackspace, and SUSE.

Modular architecture components

OpenStack has a modular architecture that includes three components as part of the effort to standardize the IaaS. Each is given a code name.

- Compute (Nova): Provides open source software and standards for large-scale deployments of automatically provisioned virtual compute instances.
- Object Storage (Swift): Provides open source software and standards for large-scale, redundant storage of static objects.
- Networking (Neuron): Provides "networking as a service" between network interface devices (vNICs) managed by other OpenStack services, such as Nova.

Shared services

OpenStack has several shared services that span the three pillars of compute, storage, and networking. These services include:

- **Identity** to provide unified authentication across all OpenStack projects
- **Image** to provide delivery services for virtual disk images
- **Telemetry** to provide aggregated usage and performance data across the services deployed in an OpenStack cloud
- **Orchestration** to provide a template-driven engine that allows application developers to describe and automate the deployment of infrastructure

- **Dashboard** to provide a web-based user interface to OpenStack services including Nova and Swift

They all integrate the OpenStack components with one another.

Neuron and SDN

Neuron, the codename for networking services, could be used to optimize IaaS with SDN. Network administrators can take advantage of SDN technology like OpenFlow to allow for high levels of multi-tenancy and massive scale of moving data from one network device to another. (OpenFlow is an open source standard, communications protocol that provides over-the-network access to a network switch's or router's forwarding plane so the remote controllers can determine the path of network packets through the network of switches.)

Users can create their own networks, control traffic, and connect servers and devices to one or more networks also through SDN technology.

Neuron has an extension framework allowing additional network services, such as intrusion detection systems (IDS), load balancing, firewalls, and virtual private networks (VPN) to be deployed and managed.

Meeting expectations: OpenDayLight

Neuron with SDN technology using the OpenFlow standard is not enough. Better suited for optimizing the IaaS is OpenDayLight, a Linux Foundation project. It's an open source framework that takes advantage of SDN and the NFV controller that administrators can use with OpenStack Neuron to help users take proactive steps in optimizing the IaaS.

Let's take a closer look at what OpenDayLight is about and what it has been doing to standardize the IaaS.

The SDN and NFV controller is contained within its own Java™ Virtual Machine (JVM). This means OpenDayLight isn't just for Linux®; it's an open platform designed to be used on any hardware and operating systems that support Java code.

OpenDayLight was founded by 18 companies, including Cisco, Dell, Juniper, IBM, and Intel; the project now counts 36 members. More than 150 developers actively contributed to Hydrogen, the first release of OpenDayLight.

Hydrogen

Hydrogen is available to enterprises, service providers, equipment providers, and academia. It comes in three editions in one package:

- The Base Edition to run on a laptop to connect to a testing tool that provides a synthetic network.
- The Virtualization Edition to add data center virtualization technologies. It is built on the base edition.

- The Service Provider Edition to help service providers and carriers develop a plan to migrate to SDN and NFV as well as support for traffic engineering. It has SNMP protocol support and APIs to manage legacy network equipment.

Network Function Virtualization

As I explained earlier, NFV decouples the network functions from proprietary hardware appliances, so the functions can run in software. It's designed to consolidate the networking components needed to support a fully virtualized infrastructure, including virtual servers, storage, and even other networks. It utilizes standard IT virtualization technologies that run on high-volume service and storage hardware to virtualize network functions.

Let's take a look how NFV works with SDN. From OpenDayLight's high-level view, SDN is described in three layers.

- **Network Apps and Orchestration:** The top layer consists of business and network logic applications that control and monitor network behavior. They include orchestration applications that are needed to globally control network traffic.
- **Controller Platform:** The middle layer stands between the SDN's northbound and southbound interfaces. The northbound interface provides a set of common APIs to the application layer. It connects with the southbound interface implements one or more protocols (such as OpenFlow) for control of the physical hardware within the network.
- **Physical and Virtual Network Devices:** The bottom layer consists of the physical and virtual devices, switches, routers, and so on that make up the connections between all endpoints within the network.

Controller risk mitigation

SDN does come with vulnerabilities (such as SDN controller hacking) that can be exploited by hackers. To mitigate controller risks, you should follow these four steps.

- Identify assets.
- Identify vulnerabilities and threats.
- Assess risks.
- Fix with safeguards.

Identify assets

Start the risk-mitigation process by identifying assets associated with the controller. Determine the categories the assets should belong to; here are some examples:

- Hardware: Network devices and switches and SDN administrator's console
- Security: Encryption mechanisms, security testing tools, and firewalls
- Administration: OpenStack and OpenDayLight guidelines
- Documentation: SDN administrator's point of contact, training manuals, network standards, disaster recovery plans, and Service Level Agreement

Identify vulnerabilities and threats

Hackers are not the only threat agents who could take advantage of the controller vulnerabilities. Another possible threat agent includes the SDN administrator, who could improperly configure the controller (and the firewalls).

Improper configuration of the controller and firewalls could lead to IaaS outages and attacks. If the policy on the IaaS failover mechanism from one region to another is not in place, the users could go to a different IaaS host provider.

Assess risks

The users want to be assured of continuous IaaS interoperability and availability and that their demand for more traffic can be met through IaaS optimization. One method of assessing risks of IaaS unavailability is quantitative. Some examples include:

- Estimated frequency that the IaaS would become unavailable
- Estimated frequency of network attacks due to improper controller configuration
- Estimated frequency of not meeting performance guarantees set forth in a Service Level Agreement
- Estimated frequency of failed failover of network routers and switches

Fix with safeguards

Cost-effective safeguards are one way of mitigating controller risks. The SDN administrator should ensure that:

- The controller is properly configured and secured by at least auditing who has accessed it, encrypting the traffic and activating the logging option.
- Network services are in place to block network attacks. They include intrusion detection systems (IDS), load balancers, and firewalls.
- Failover mechanism is in place to fail over quickly from unhealthy network routers and switches to healthy ones.
- SDN/NFV administrator has the proper skills and instruction to manage the system.

Conclusion

In planning for optimizing IaaS with SDN, consider the best practices for resolving IaaS interoperability issues and setting up an IaaS cloud service model. Taking proactive steps on how OpenStack-based IaaS can be used with OpenDayLight to optimize the IaaS should be part of the plan. Make sure that the controller risk mitigation plan is in place. You need to build a team of IaaS cloud service players, managers, business analysts, and system engineers and make it easier for them to do their job of optimizing IaaS with SDN.