Transforming to DevOps with Junos OS

Innovations Facilitate a Rapid, Continuous-Delivery DevOps Environment for Service Creation and Delivery

Table of Contents

Executive Summary

Traditional information technology service management (ITSM) methods establish disparate test and production systems that often consume vast operational resources while adding an equal amount of complexity during the service creation and upkeep process. DevOps is the unification of systems *developers* and *operators* use for continuous delivery of services with higher quality and faster service creation, at a lower price.

This white paper is for network systems developers and operators who wish to transform operations models for their Juniper-based Internet Protocol (IP) networking infrastructures using IT DevOps methodologies. The paper will examine how specific features and capabilities of Juniper Networks® Junos® operating system, the OS that powers all Juniper routers, switches, and firewalls, not only integrate into existing IT DevOps environments but also help build and maintain networks for continuous delivery—a key concept of DevOps.

Introduction

The ability to deliver highly available networking services is an absolute requirement. Many mission-critical applications rely on the network infrastructure as the primary service delivery vehicle. Network operators and systems administrators alike are challenged to maintain infrastructures with 100% availability and zero downtime; the network must be available at all times and at all costs, even during service-related upgrades or maintenance operations following a failure. In many instances, preserving uptime and availability means new applications and services must be tested and moved into production without impacting existing services. How is this possible? Is it feasible to build redundancy at every stage of the network, from the end user to the application server? Is there a better way?

DevOps methodologies have addressed some of these challenges by exploiting automation and orchestration across all elements of IT infrastructures—compute, storage, and networking. Automation helps eliminate repeatable manual tasks through scripts or other software tools; orchestration is an extension of automation that groups automated tasks into workflows. In the realm of IT infrastructures, these workflows may include several networking, server, and storage devices. As a result, these automation tools not only have to be agile and robust enough to efficiently control the underlying devices, they must also be versatile and open enough to communicate securely through a rich set of programmatic interfaces and protocols for seamless integration into frameworks that comprise the complex ecosystem of orchestration tools.

Among Juniper's product offerings, Junos OS provides components that are critical to this ecosystem, delivering automation tools that execute repeatable manual tasks at speed, reducing errors, and achieving scale for service delivery while integrating seamlessly with popular orchestration frameworks used in established DevOps IT environments.

DevOps with Junos OS

The most common application of automation tools is in the management and provisioning of IT resources. IT management, including the network, is grouped by the International Standards Organization (ISO) into a framework called FCAPS, which stands for Fault, Configuration, Accounting, Performance, and Security. Traditionally, functions and tasks in each of these management categories are executed by a collection of ad hoc scripts, tools, or management products that help systems integrators and operators efficiently deploy, monitor, and manage IT infrastructures.

Automation as it relates to network devices and services can also be categorized into three areas, with reference to how and when they are utilized in the IT lifecycle. First, day-zero tools help set up, configure, and provision devices as they are deployed for the first time. Second, day-to-day tools help automate performance monitoring activities of devices while providing consistency and accuracy, often at scale. Last but not least, on-demand tools provide an automated means for applying configuration changes when provisioning IT resources and services, or fault management when failures occur on an on-going or as-needed basis. All three areas are equally important to consumers of IT technology, whether they are end-user subscribers or service(s) providers, as these tools bring new network-dependent services to market quickly and keep them operational with low, if not zero, downtime.

Another way to look at the importance of automation and orchestration in IT environments is how they relate to the growth and wide adoption of virtualization, which has propelled the need to control and manage compute, storage, and network resources by a single orchestration framework. Service providers often use such orchestration systems to integrate self-service portals, consumption-based metering, chargeback systems, and the creation of service catalogues.

Emerging trends show that operations and business support systems (OSS/BSS) are tightly integrated with underlying physical and virtual hardware infrastructures by way of these automation and orchestrations tools to deliver services that can be monetized and delivered by the push of a button. Since these services are so dependent on the underlying network infrastructure, either as the service itself—as in the case of communications service providers (CSPs) and cable operators—or as the delivery mechanism for Web 2.0, cloud-based (computing, storage, or application) vendors, the ability to reliably provision and manage networking resources with speed and at scale defines success or failure in these emerging markets. This requires an efficient and reliable way to deliver IT services quickly, efficiently, and with no time to pass blame when problems arise. In other words, it requires DevOps.

What Is DevOps?

DevOps is a software or systems development methodology that stresses communication, collaboration, and integration between software or systems developers and implementers such as IT operations professionals. DevOps methods encourage rapid development and deployment of software or systems with an emphasis on continuous improvement of quality, security, and reliability through rapid changes delivered via automation as a vehicle for easing the pain of the resulting dynamic environment. DevOps deviates from traditional software and systems development methodologies in that it changes software functionality frequently to quickly move improvements into production. While frequent and interactive changes in an IT environment can be detrimental to services and quality at many levels, automation and orchestration tools reduce if not eliminate their negative impact.

Effective DevOps requires a culture and mindset of collaboration between developers and operations. For IT environments, this is inclusive of compute, storage, and network systems administrators and operators. Rapid service creation and delivery from existing or new infrastructure requires expedited incubation, design, planning, development, implementation, test, delivery, and operations. In the operations phase, continuous monitoring and improvements are subject to this same cycle; hence the term "continuous delivery," which is often used to describe the DevOps model.

The Challenges

Integration is key. And, the ability to create new services on an existing infrastructure without bringing down existing services is vital. How is this possible? DevOps methodologies carve a test system out of the production environment; once new services are developed and tested on this test system, they are simply rolled into production. With developers and operators working side by side, using software tools to *orchestrate* the creation and deployment of new services, IT shops have finally figured out a meaningful and productive way of achieving success. However, there are plenty of other challenges that remain.

DevOps for NetOps?

The transformation of DevOps practices into NetOps is not as straightforward as the adoption of these practices in the virtualized server environment. The rigid operational environments and resource limitations on network devices make it much harder to carve out a test network from a production network than it is to spin up test virtual machines (VMs) on a production compute server. Despite these types of challenges, NetOps adopters have embraced DevOps methodologies. An IT environment that can automate and orchestrate everything but the network would not be quite as efficient as an environment that could orchestrate the entire infrastructure.

IT automation and deployment frameworks used in DevOps, such as Chef, Puppet, and Ansible, have naturally been extended to support networking products, since the network is integral to these IT infrastructures. Some of these frameworks, namely Chef and Puppet, require a client agent in the managed device; others, such as Ansible, are clientless mechanisms that utilize protocols, APIs, or CLIs to communicate with the target device. Many existing legacy devices may not support a simple interface into the automation framework. Thus, the goal is not just to automate the configuration and provisioning of one network device, but to provision the end-to-end network service as part of the orchestrated delivery of an IT service using a turnkey mechanism, effective across multiple devices including physical or virtual compute, storage, and network resources. Software-defined networking (SDN) and Network Functions Virtualization (NFV) also play key roles in this new paradigm of service creation and delivery through automation and orchestration.

Even tools for deploying new hardware and software features in routers, switches, and firewalls without incurring any downtime can help NetOps maintain network systems with higher availability. Software updates, real-time configuration changes, and the introduction of new hardware in modular platforms could cause heavy network outages if the device, network, or IT framework is not designed for high availability. Fast-fail and recovery through fault isolation and rollback are preferred by IT operators during changes; unfortunately, not all devices support these mechanisms.

Developing automation and orchestration solutions also costs money. Unless there is a compelling business reason to develop the scripts, tools, and widgets required to deliver push-button service delivery, IT shops will typically opt to use the same old time-tested and error-prone manual methods to deliver IT solutions.

The Solutions

Where there is a business case to provide nonstop services, high availability, and zero tolerance for downtime, IT solutions that self-heal in times of failure are typically enabled by *automation* and *orchestration*. The truth is, if something can break, it will, and that could be any component in the IT infrastructure, including the network.

The key to success is rapid problem isolation and recovery. In some cases, this means looking for evidence of an impending problem and proactively taking remedial steps to avert disaster and downtime before the problem actually occurs. Since there are many moving parts in a dynamic IT infrastructure, such a solution must be robust and able to adapt to dynamic workloads and operate at *scale*. Solutions that can *create* services and adapt to new problems must be built on programmable devices that can be dynamically changed without affecting existing services.

Programmability requires an extensible operating system such as Junos OS, which provides an agile and open operating environment that can be adapted to deliver IT services not only on a box-by-box basis, but from a systems level. Junos OS provides the building blocks required to build robust networks that can be programmed, monitored, and managed as a whole, creating a foundation for the overarching network resource that is vital to the creation and delivery of IT services that can be monetized.

vMX and the Contrail Cloud Solution for NFV

In the context of DevOps, NFV provides a flexible and agile way not only to create services, but also to create and test solutions without expensive hardware investments. This is a game-changing paradigm shift that promotes the virtues of DevOps. A virtualized version of Juniper Networks MX Series 3D Universal Edge Routers is a prime example of an NFV platform that can be verified before moving into production on real hardware or in virtualized production environments. The Junos OS-based Juniper Networks vMX 3D Universal Edge Router is a full-featured software-based implementation of MX Series routers that run on virtualized x86 hypervisor platforms. DevOps teams can now create and test solutions utilizing the automation frameworks discussed later in this paper in a completely virtualized environment. The Juniper Networks Contrail Cloud Platform architecture can also be used to build and orchestrate vMX-based solutions on Contrail OpenStack, a distribution of OpenStack software from Juniper Networks.

Junos Continuity

Junos Continuity is another game changer for DevOps practices in the networking space. Junos Continuity is an enhancement to Junos OS designed to allow new hardware or upgrades to be made without having to reboot or restart a device, avoiding costly downtime. The Junos Continuity feature allows users to add hardware in modular chassis, even with support for new ASICs, by simply loading device drivers and support software to enable the new hardware without disrupting operations. An industry first, this feature exploits the modularity of the Junos OS architecture to its full potential.

The biggest benefit of this feature, especially for service providers, is avoiding hefty requalification cycles when making simple feature or interface additions. Whereas previously service providers had to spend months qualifying a software release just to incorporate a new line card or I/O module into a modular chassis, they can now perform integration tests on the existing software release and verify that the hardware works in a matter of weeks. This is a huge savings, and helps deliver new features and services to market faster.

When it comes to supporting DevOps for NetOps capabilities, however, Junos OS does not stop there.

The Automation and Orchestration Ecosystem from Juniper Networks

Figure 1 summarizes the product families and tools incorporated into Juniper's growing automation ecosystem, which is constantly adding new protocols, APIs, and client tools to incorporate new scripting and programming languages in favor of orchestration tools.

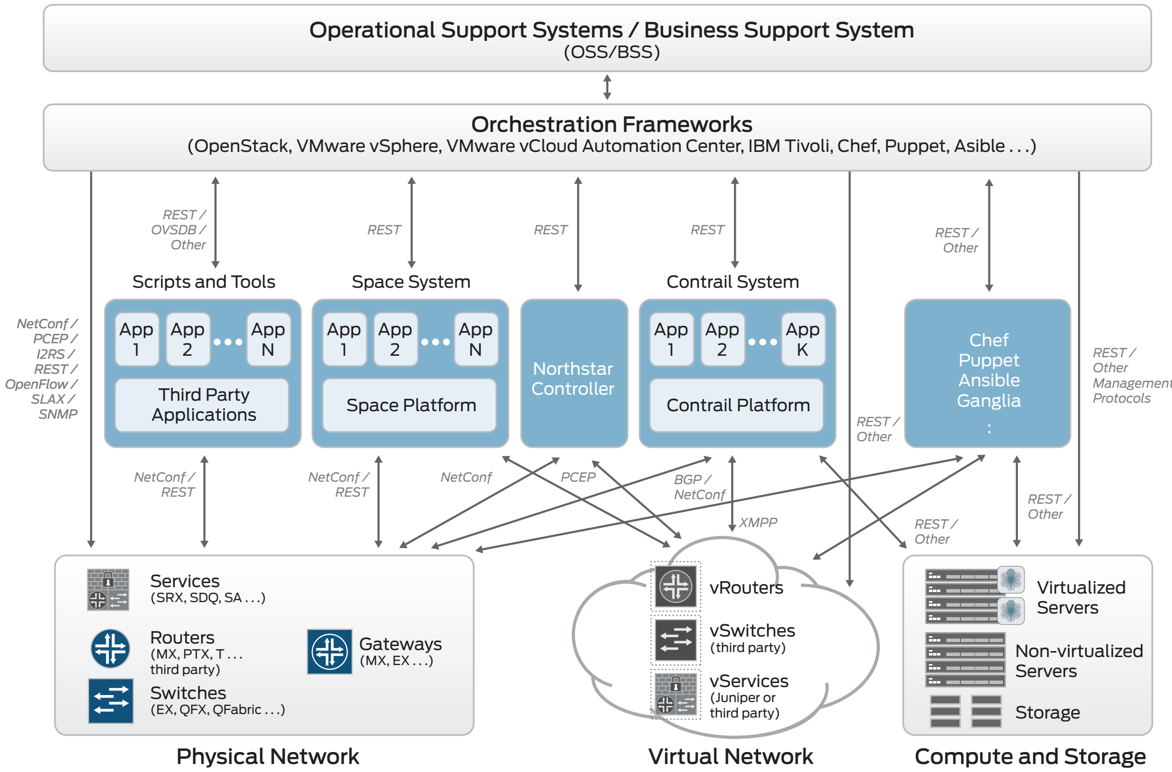

Figure 1: Juniper Networks automation and orchestration ecosystem

As shown in Figure 1, Junos OS, which runs on Juniper Networks physical and virtual switches, routers, and firewalls, supports many different protocols and APIs that can be adapted for FCAPS management.

At the top are OSS/BSS, which are usually custom-built applications implementing business procedures, policies, and operations. These systems communicate with one or many orchestration frameworks or management applications via protocols and/or APIs. These orchestration frameworks are often the intermediate layer that aggregates southbound configuration requests, or northbound events and statistics correlating activity across multiple platforms. South of the orchestration frameworks are the management and controller applications—for example, the Juniper Networks Junos Space Network Management Platform. Junos Space supports a number of management applications, including:

- Network Director, which provides a comprehensive view of the physical and virtual network through a single pane of glass. Network Director also integrates with third-party orchestration tools such as OpenStack, VMware vCloud, and vSphere.
- Security Director, which performs policy management of virtual and physical Junos OS-based firewalls.
- Services Activation Director, which is a suite of products that provides network provisioning and management functions that include:
  - Network provisions of L2/L3 VPNs across MPLS and carrier Ethernet
  - Transport used for design, provisioning, and activation of RSVP-signaled and static label-switched paths (LSPs)
  - QoS for creating quality-of-service (QoS) profiles for specific Ethernet services
  - SLA to measure network performance using Y.1731 and RFC2544, and perform fault management using Ethernet connectivity fault management (CFM), Ethernet link-level fault detection and management, and Bidirectional Forwarding Detection (BFD)
  - Sync to configure and provision synchronization interfaces, including IEEE1588-2008(PTP) and Synchronous Ethernet

These services are accessed through a northbound REST-based API that also enables network providers to tap into the Junos Space platform to build native applications for SDN architectures.

Similarly, SDN controllers such as the Juniper Networks NorthStar Controller for Junos OS-based core routers, and Contrail for the virtualized Contrail vRouter, provide traffic management for service chaining on physical and NFV platforms. These applications have interfaces for programmable control, management, and monitoring, utilizing RESTful APIs or protocols.

Third-Party Integration

Where a Juniper Networks product does not provide an essential functionality in the creation or monitoring of a robust automated network service, Junos OS enables an interface, protocol, or API, or adds a third-party client to integrate with a best-in-class product.

Puppet, Chef, and Ansible are popular automation frameworks that can be used with Junos OS-based products, allowing operations staff to integrate the network footprint with established IT frameworks and begin a transformation to DevOps.

RESTful API

A RESTful API uses HTTP or secure HTTPS requests to GET, PUT, POST, and DELETE data on a target device.

Representational State Transfer (REST), which is used by browsers, can be thought of as the language of the Internet. Now that cloud usage is on the rise, APIs are emerging to expose Web services, and REST is a logical choice for building APIs that allow end users to connect and interact with cloud services. RESTful APIs are used by many IT equipment manufacturers, cloud service providers, and online applications to extend manageability, automation, and orchestration.

Puppet

Devices running Junos OS support Puppet software for configuration management. Developed by Puppet Labs, Puppet provides an efficient and scalable software solution for managing the configurations of large numbers of devices. System administrators use Puppet to manage computer resources such as physical and virtual servers and network devices. Puppet is typically deployed using a client/server arrangement, where the server—or Puppet master—manages one or more agent nodes. The client daemon—or Puppet agent—runs on each of the managed computer resources.

The Juniper Networks “netdev” Puppet module provides new Puppet resource types for configuring:

- Physical interfaces
- L2 switch ports
- VLANs
- Link aggregation groups

When using Puppet to manage Junos OS-based devices, the Puppet agent:

- Makes configuration changes under exclusive lock
- Logs all commit operations with a Puppet catalog version for audit tracking

Puppet report logs include:

- A Junos OS source indicator for log entries specific to Junos OS processing
- Tags associated with the operation or error to enable easy report extraction

Chef

Chef is an infrastructure automation and configuration framework. The Chef client is supported on some Junos OS-based devices. Chef runs in a client/server model where predefined (recipe) templates for configurations are stored on the Chef server. Devices running the Chef client request changes from the Chef server upon boot-up and the Chef recipes are transferred to the client device. The strength of Chef is its ability to manage an entire IT infrastructure from a client workstation where DevOps professionals bundle multiple recipes into Chef cookbooks to make orchestrated configuration changes across multiple types of devices. Chef can be used to install applications as well.

The Juniper Networks Chef module provides options for configuring:

- Physical interfaces
- L2 switch ports
- VLANs
- Link aggregation groups

When using Chef to manage Junos OS-based devices, the Chef agent:

- Makes configuration changes
- Logs all commit operations

Chef report logs include:

- A Junos OS source indicator for log entries specific to Junos OS processing

Other Third-Party Tools

Since Junos OS supports standards-compliant and open protocols, APIs, and interfaces, many other third-party network management, monitoring, automation, and orchestration tools can be used to manage Juniper devices. This is critical, since DevOps teams may use several different tools due to personal preference. Open connectivity provides the greatest opportunity for seamless integration.

Splunk and Ganglia are just two examples of other third-party tools that can be used to manage and monitor Junos OS-based devices.

- **Splunk**: Splunk is a log analytics engine that can easily integrate with Juniper devices. It captures, indexes, and correlates real-time data in a searchable repository and then uses that data to generate graphs, reports, alerts, dashboards, and visualizations. Splunk makes machine data accessible across an organization and identifies data patterns, provides metrics, diagnoses problems, and provides intelligence for business operations. Splunk offers customized templates that interface with devices running Junos OS.
- **Ganglia**: Ganglia is a scalable distributed monitoring system for high-performance computing systems and network devices. It leverages widely used technologies such as XML for data representation; XDR for compact, portable data transport; and RRDtool for data storage and visualization. Junos OS-based devices can be monitored using Ganglia.

Automation Tools in Junos OS

Junos OS automation consists of a suite of tools used to automate operational and configuration tasks on Junos OS-based network devices. These tools leverage the native XML capabilities of Junos OS and include commit scripts, operation (op) scripts, event policies, event scripts, and macros for ***on-box automation***.

Junos OS automation simplifies complex configurations and reduces potential configuration errors, saving time by automating repetitious operational and configuration tasks. It also speeds troubleshooting and maximizes network uptime by warning of potential problems and automatically responding to system events. Junos OS automation can also capture the knowledge and expertise of experienced network operators and administrators, allowing a business to leverage this combined expertise across the organization—all potential DevOps benefits.

Junos OS automation scripts can be written in one of the following scripting languages:

- **Extensible Stylesheet Language Transformations (XSLT):** XSLT, a standard for processing XML data, is designed to convert one XML document into another.
- **Stylesheet Language Alternative Syntax (SLAX):** An alternative to XSLT, SLAX has a simple syntax that follows the style of C and PERL but retains the semantics of XSLT. Programmers familiar with C often find it easier to learn and use SLAX. Scripts written in one language are easily converted to the other.

The following sections describe the different types of functionality for Junos OS automation.

Commit Scripts

To automate the commit process, Junos OS configuration automation uses commit scripts that enforce custom configuration rules. When a candidate configuration is committed, it is inspected by each active commit script. If a configuration violates a custom rule, the script can instruct Junos OS to take one of the following actions:

- Generate and display a custom warning message to the user
- Generate and log custom system log (syslog) messages
- Change the configuration to conform to the custom configuration rules
- Generate a commit error and halt the commit operation

When used in conjunction with macros, commit scripts simplify the Junos OS configuration process while extending it with custom configuration syntax.

Operational Scripts

Junos OS operations automation uses op scripts to automate operational tasks and network troubleshooting. Op scripts can be executed manually in the CLI or upon user login. They can also be called from another script. Op scripts can process user arguments and can be constructed to:

- Create custom operational mode commands
- Execute a series of operational mode commands
- Customize the output of operational mode commands
- Shorten troubleshooting time by gathering operational information and iteratively narrowing down the cause of a network problem
- Perform controlled configuration changes
- Monitor the overall status of a device by creating a general operation script that periodically checks network warning parameters, such as high CPU usage

Op scripts can be used effectively to streamline DevOps environments.

Event Scripts and Event Policy

Junos OS event automation uses event policy and event scripts to instruct the operating system to perform actions in response to system events.

Event Policy: An event policy is an if-then-else construct that defines actions to be executed by the software on receipt of an event such as a system log message or SNMP trap. Event policies can be executed in response to a single system event or to correlated system events. Multiple actions can be configured for each policy, including the following:

- Ignore the event
- Upload a file to a specified destination
- Execute Junos OS operational mode commands
- Execute Junos OS event scripts

Event Scripts: Event scripts are triggered automatically by defined event policies in response to a system event, and can instruct Junos OS to take immediate action. An event script automates network troubleshooting and network management by doing the following:

- Automatically diagnose and fix problems in the network
- Monitor the overall status of a device
- Run automatically as part of an event policy that detects periodic error conditions
- Change the configuration in response to a problem

Commit scripts, op scripts, and event scripts can all be utilized for DevOps in conjunction with off-device DevOps frameworks such as Puppet, Chef, and Ansible. Typically, these methods are combined with zero touch provisioning (ZTP) used for provisioning and configuring devices on day-zero deployments.

Junos SDK

The Junos software development kit (Junos SDK) lets developers innovate on top of Junos OS by building and validating new and existing network applications. Though the scripts discussed previously are on device automation tools, the Junos SDK provides ***off-device automation*** capabilities. This is particularly important to DevOps as the open APIs provided with the Junos SDK let applications run on Junos OS-based devices with remote access to the control, services, and virtual planes. The Junos SDK transforms Junos OS from a proven, carrier-grade, high-performance network operating system into an application platform that can be used by developers with common tools and APIs for rapid network application development across routers, switches, and firewalls. The Junos SDK exposes platform and system functions without compromise, while enhancing the robustness and security that have always been integral to Junos OS. The remote control and flexibility to run network applications remotely is highly beneficial to DevOps methodologies.

Juniper Networks Junos SDK applications include the following components:

- **Control plane programs:** Central control-style applications can access Junos OS system state, as well as routing, switching, and security features from a Routing Engine.
- **User interface extensions:** Plug-ins seamlessly extend the UI schema for commands and the configuration database. Operators can administer apps with new syntax through any Junos OS interface such as the CLI, NETCONF, Junos Space, or Junos Automation. SNMP and system logging extensions are also feasible.
- **Service plane programs:** Service plane programs operate with many real-time POSIX threads, each executing a packet polling loop. They are granted exclusive use of an entire multiprocessing services engine on a services hardware module. Applications are generally optimized for low latency and high throughput for their important task of receiving, processing, and sending packets.
- **Service plane plug-ins:** Service plane plug-ins are also components that can access packets at line speeds. Their packet, event-driven model allows for application chaining within a single services engine. These components can be developed more quickly by taking advantage of generic session management infrastructure and APIs.
- **Virtual plane programs:** Virtual plane programs execute in an on- or off-device system and can virtually integrate with Junos OS using the remote routing and system APIs.

NETCONF Protocol

Network Configuration Protocol is a network management protocol which provides mechanisms to install, manipulate, and delete the configuration of network devices. Its operations are realized on top of a simple remote procedure call (RPC) layer. The NETCONF protocol uses Extensible Markup Language (XML)-based data encoding for the configuration data as well as the protocol messages. The protocol messages are exchanged on top of a secure transport protocol.

Junos SDK supports the following major API categories:

- **Routing and system APIs:** APIs for the device-local Junos OS control plane-based apps to use and manipulate routing, filtering, high availability, interface state, and other Junos OS subsystems in dynamic ways, including APIs and languages to extend the Junos OS user interface.
- **Remote routing and system APIs:** Environment- and language-agnostic middleware renditions of routing and system APIs for device-local apps on Junos OS, and remote virtual plane-based apps like those that are virtually hosted, and running in non-Junos OS operating systems like Linux. Within the Junos OS service model, a virtual plane extension to the services plane can run compute-intensive traffic processing or other control plane-style network applications.
- **Traffic services APIs:** APIs for the device-local Junos OS services plane-based apps accessing selected packet streams from the data (forwarding) plane for transforming and monitoring purposes, including zero-copy packet manipulation APIs with line-rate potential traffic processing on Juniper Networks multi-services and application services line cards.
- **Junos XML management protocol API:** Uses Junos XML protocol Perl client modules to develop custom applications for configuring information on devices that run Junos OS. Client applications use the Junos XML management protocol to request and change configuration information on Juniper Networks routers. The Junos XML management protocol is customized for Junos OS, and operations in the API are equivalent to those in the Junos OS CLI.
- **NETCONF application programming interface API:** Uses NETCONF Perl client modules to develop custom applications for configuring information on devices that run Junos OS. Client applications use the NETCONF XML management protocol to request and change configuration information. The NETCONF XML management protocol includes features that accommodate the configuration data models of multiple vendors.

The Junos SDK provides the ability to build on-box automation and orchestration tools by creating a network-wide application platform built on the foundation of programmability offered by the Junos OS. DevOps methods can be extended beyond the traditional automation and orchestration frameworks with this type of programmable network application environment.

Conclusion

Plain and simple, network agility has to match business agility. Specifically, the network must integrate with the evolving business landscape and the already evolving IT DevOps agility advancements. A strategy that unifies IT and IP network domains is vital for service providers to transform their business and operations. Automation and orchestration help service providers, implementers, operators, and, in some cases, even end users deal with their challenges and underlying problems. Since these solutions are constantly evolving, as are the tools for automation and orchestration, the networking infrastructure has to adapt to new hardware and software capabilities over time without paying a penalty for downtime of existing services. The Junos OS and its rich set of capabilities for supporting automation and orchestration help service providers deliver a solid networking infrastructure for DevOps methodologies being widely adopted within IT services delivery.

About Juniper Networks

Juniper Networks is in the business of network innovation. From devices to data centers, from consumers to cloud providers, Juniper Networks delivers the software, silicon and systems that transform the experience and economics of networking. The company serves customers and partners worldwide. Additional information can be found at www.juniper.net.

Corporate and Sales Headquarters
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089 USA
Phone: 888.JUNIPER (888.586.4737)
or +1.408.745.2000
Fax: +1.408.745.2100
www.juniper.net

APAC and EMEA Headquarters
Juniper Networks International B.V.
Boeing Avenue 240
1119 PZ Schiphol-Rijk
Amsterdam, The Netherlands
Phone: +31.0.207.125.700
Fax: +31.0.207.125.701

2000586-002-EN Oct 2015