Best Networking Practices in a DevOps World

Published 6 May 2020 - ID G00720717 - 17 min read

By Analysts Andrew Lerner, George Spafford

Initiatives: Agile and DevOps

In theory, DevOps includes all aspects of infrastructure; but in practice, networking is largely disconnected from DevOps initiatives, particularly within on-premises data centers. This research identifies how I&O leaders should integrate networking activities into DevOps activities.

Overview

Key Challenges

- Many I&O leaders are unable to evolve infrastructure to meet release delivery cadence demands for DevOps initiatives.
- More than 65% of enterprise networking operational activities are performed manually, which constrains rapid delivery of infrastructure.
- Networking teams have limited involvement in DevOps initiatives, which can limit the effectiveness of DevOps projects, including delaying provisioning/deployments.

Recommendations

Infrastructure and operations (I&O) leaders focused on enabling agile and DevOps practices must work with networking teams to:

- Remove network-related delivery impediments to DevOps initiatives by embedding networking subject matter experts (SMEs) into DevOps product teams — when and where appropriate.
- Evolve training and incentive plans for networking personnel by reskilling with a focus on automation, and by creating agility-focused KPIs.
- Integrate networking operations into the DevOps toolchain by managing network configurations as code to improve infrastructure agility.

Strategic Planning Assumptions

By 2023, 15% of on-premises data center networking activities will be integrated into a CI/CD pipeline, an increase of more than 15 times from early 2020.

By 2023, 60% of data center networking activities will be automated, up from 30% in early 2020.
Introduction

As organizations implement digital initiatives, reducing time to deliver business outcomes is critical. Organizations look to DevOps practices to improve the speed of application delivery while better accommodating the uncertainty of application and infrastructure requirements. However, roughly one-third of these organizations have not yet achieved their agility goals. A key reason for this, noted by clients during inquiry, is “hitting the wall requesting servers.” This issue is less pervasive in the public cloud, where automation is natively supported; but it is particularly challenging on-premises. In these scenarios, it often boils down to one of several key constraints: networking, firewall and/or database. Similarly, I&O leaders report the top challenge to scaling DevOps efforts is managing legacy systems, and many DevOps teams refer to the network as a legacy system.

While Gartner fielded more than 5,000 client inquiries on DevOps from February 2019 through February 2020, networking was discussed in only 4% to 5% of these calls (see Figure 1) compared to other DevOps topics (such as agile, cloud, tools and automation).

![Figure 1. Key Topics Discussed in Gartner DevOps Inquiries](chart)

The goal of DevOps is to improve the flow of customer value. This is accomplished by identifying and removing the greatest constraint to flow with each cycle of improvement (see “6 Steps to Increase DevOps Release Velocity by Removing Constraints”). Given that networking is a
substantial constraint in many DevOps initiatives (often hidden behind server provisioning), this research identifies how to better integrate networking practices with DevOps initiatives.

**Analysis**

**Embed Networking Personnel Into DevOps Teams**

DevOps teams are usually composed of six to 10 individuals, including developers, quality assurance, infrastructure engineers and product managers. However, networking SMEs are often underrepresented in these teams. Thus, networking is a major constraint to on-premises application delivery and undermines agility.

Thus, we recommend inviting networking SMEs to attend stand-ups, sprint planning and retrospectives. Further, when networking is the greatest constraint, I&O leaders should embed networking personnel into DevOps product teams. These networking personnel are tasked with identifying where the greatest network constraint exists (such as switch port provisioning, access control lists [ACLs], firewall rules, load balancer configuration, change management, approval, testing and validation) and reducing it. Further, an embedded network engineer will enable the DevOps team to understand network activities/constraints, and the network team to better understand the needs/asks of the DevOps teams.

While I&O leaders do not necessarily have to embed their most senior or skilled network personnel, they should engage someone who has:

- A solid understanding of network configurations, troubleshooting and change control processes across routers, switches, load balancers and firewalls
- Authorization to make network changes
- A willingness to learn
- A collaborative mindset, including a willingness to share knowledge
- Familiarity with network automation tools and concepts

However, DevOps teams cannot scale if they need one of everybody on every team. Not all DevOps product teams will require networking expertise. Some teams may only require networking personnel for specific tasks (not for the full duration of the project). As the network team becomes more familiar with both the DevOps means of working and their common needs, adoption of network automation will allow for better scaling of network participation. For additional insight on organizing and staffing DevOps teams, see “How to Scale DevOps by Building Platform Teams.”

**Adjust Network Team KPIs and Reskill Network Teams With a Focus on Network Automation**
I&O leaders must change how they motivate, measure and reward networking personnel. Successful DevOps initiatives require all stakeholders to have shared, business-oriented objectives and metrics. However, most network teams are measured and compensated based on uptime-focused metrics, which have encouraged them to successfully pursue incremental infrastructure modernization for the past decade. To break out of this incremental approach, I&O leaders must facilitate a key shift in how networking personnel are measured and compensated.

We recommend introducing new KPIs that reflect a broader service orientation, with a particular emphasis on increasing agility (see “Adopt I&O Key Performance Indicators to Deliver Digital Business Value”). Examples of agility-focused KPIs include:

- Mean time to service delivery
- Mean time to detection of outage
- Percentage of network changes automated
- Percentage of network operational activities automated
- Percentage of manual tasks eliminated

These new performance metrics must serve as the foundation for how networking personnel are rewarded and compensated.

Focus on Automation and Programmability

To best support a DevOps culture, networking personnel need expertise in network automation and network programmability. For example, this includes Python, Ansible, Linux, APIs, and cloud provider networking capabilities (see Note 1 and “Jump-Start Network Automation to Scale Digital Initiatives”). Unfortunately, there is limited automation in most enterprise networks today (see Figure 2).

Figure 2. Primary Mechanism for Network Changes
Networking personnel with midlevel or advanced expertise in these areas are scarce and command a substantial salary premium. We believe a lack of time allocated to self-paced learning is driving the network automation skills shortage.

Thus, I&O leaders must facilitate a revamp or “upskill” of network personnel by allocating time for learning these new skills. More than half of networking personnel who have automation expertise had to develop these skills on their own time. To get started, I&O leaders responsible for networking and DevOps should take the following five simple actions:

1. Have networking team members take an introduction to Python training class. This entry-level course can be completed in one day or less. In addition, AppDev teams could hold a brown bag session with network teams on basic Python and coding concepts.

2. Have networking team members download popular network-friendly DevOps tools and spend one to two hours exploring and gaining familiarity with these tools (see Note 1 for a sample list of tools).

3. Create a shared repository of networking tasks that are performed manually, but that have the potential to be automated. Focus specifically on tasks that are the greatest constraint to DevOps teams.

---

**Primary Mechanism for Network Changes**

- **CLI on Individual Devices**: 71% in 2017, 70% in 2018, 65% in 2019
- **Network Automation Tool**: 6% in 2017, 16% in 2018, 10% in 2019

Source: Gartner IQCS Poll (December 2017-2019)

720717_C
4. Sponsor a network automation “hackathon,” where networking personnel work with DevOps teams and software engineers in a sprint to automate a manual activity. Hackathons could involve automating the troubleshooting of a common issue (i.e., “branch X is slow” or “can you check the firewall?”).

5. Rotate network engineers to observe DevOps scrum teams to build understanding of the processes and work methods, including continuous integration/continuous delivery (CI/CD) software processes. This should provide networking personnel with a basic understanding of these processes; avoid in-depth discussions about tools or other details.

While the above activities require minimal time and cost, these initial steps are instrumental for providing networking personnel with the resources to develop critical skills. Further, there are numerous online vendor-driven communities that provide tools and can foster these types of activities, including Network Automation Heroes, Cisco DevNet, Network to Code on Slack, NRE Labs and others.

I&O leaders should also expand the career progression opportunities available to networking personnel. In addition to the traditional progression of getting deeper into networking or moving into management roles, I&O leaders should enable network team members to pursue broader career paths that offer financial and technological benefits. These nonlinear career progressions include key DevOps roles such as site reliability engineers and automation architects (see Note 2).

Automate Networks and Build Smaller Pods

Network automation is essential to removing constraints that hamper DevOps initiatives. In most cases, I&O leaders have ample opportunities to increase the level of network automation in their enterprise networks. On average, enterprises have automated only 30% of network changes.¹ I&O leaders must drive automation by default in their network by mandating that all new network purchases include automation and programmability. All new network purchases should mandate:

- Turnkey integration with tools such as Ansible, Puppet and Python, including commercial support for these integrations at both the device and controller level.

- Support for open, published, versioned, well-documented RESTful APIs, with a preference toward standardized or popular frameworks (including NAPALM, Nornir and NETCONF) and software development kits (SDKs).

- Commercially supported turnkey plug-ins for common cloud, virtualization and container orchestration, such as Kubernetes, Microsoft, OpenStack and VMware.

Support for these capabilities should be included as a requirement in all RFPs from network vendors. Prior to making a purchase, we recommend performing a proof of concept (POC) to validate usability and functionality in their specific environment with their versions.

Use Pod-Based Designs to “Fail Small”
A common DevOps mantra is to “fail fast and iterate.” However, network failures aren’t tolerated in practice in the enterprise — people get fired, and highly inefficient policies are mandated as a result. The network reality is:

- Fail very small
- Detect it immediately
- Fix it very very fast
- Don’t ever fail the same way again

Specifically, this means that networking teams build out their data center networks in smaller increments. This is a departure from how most network infrastructures are currently built, where large refresh activities occur once every several years. For example, instead of building out an entire data center of 2,500 ports at once, networking teams should build out data center pods with 100 to 400 ports each. A central backbone that provides basic IP transport should connect the pods, which are largely self-contained. This approach provides several benefits, including:

- Supports fit-for-purpose networks or zones within a data center
- Increases network resiliency by limiting the blast radius of an outage
- Helps to more quickly introduce innovation
- Reduces risk of hardware or software upgrades

Each generation of pod can copy successful characteristics of prior pods while iteratively fixing issues. This style of build-out in the data center is widely used in large-scale data centers. Most cloud providers use network pods, as it is the only way to sustain high growth rates and high availability (see “Look Beyond Network Vendors for Network Innovation”).

Integrate Networking Operations Into the DevOps Toolchain

The DevOps toolchain is a set or combination of synergistic tools enabling creation, delivery and management throughout the software product life cycle that includes several activities: plan, create, verify, preproduction, release, configure and monitor. This toolchain is not a sequential set of steps, because at any point in any of the phases another phase may be executed (see Figure 3 and “How to Build and Evolve Your DevOps Toolchains”).

Figure 3. The DevOps Toolchain
Gartner, Inc. | 720717

Manage Network Configurations as Code

Only a very small percentage (1% to 2%) of Gartner clients have successfully integrated networking activities into the DevOps or CI/CD pipeline. However, from these successes, we observe a common pattern: they manage network configurations as code. Thus, to integrate networking with the DevOps toolchain or a CI/CD pipeline, I&O leaders must *treat networking configuration changes and operational tasks as code*. Thus, network configurations and operational tasks should be managed like any other software component — this is a significant shift in current operating procedures for most organizations. Thus, I&O leaders must ensure that networking teams take the following actions:

- Store network configurations in a version control repository.
- Prepare configuration rollbacks for all proposed changes.
- Apply peer configuration reviews (similar to code reviews).
- Implement configurations via the IT organization’s software development life cycle.
- Apply automated checks to validate the change and overall operating environment.

Table 1 identifies further details and examples of how to treat network changes as code within the DevOps toolchain. Also Note 1 refers to specific tooling that helps to enable these activities.

---

**Table 1: Networking Within the DevOps Toolchain**
<table>
<thead>
<tr>
<th>DevOps Toolchain Activity</th>
<th>Network Activity</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plan</td>
<td>Network upgrades — such as adding capacity — which may include devices, features/licenses and bandwidth. Currently, many product owners treat the network as a utility; but for some initiatives, the network must be upgraded. This step could also include broader planning to better align with DevOps, such as identifying automation shortcomings.</td>
<td>Determine if there is availability of switch ports in the data center for a new server farm; or if bandwidth and latency at remote locations are sufficient for a new application. Determine common tasks that are performed manually that constrain DevOps delivery, then add to a feature backlog and prioritize by the agile/DevOps team.</td>
</tr>
<tr>
<td>Create</td>
<td>Create network configurations or network activities as repeatable, programmable and executable files (such as a text file of commands, script or an API call).</td>
<td>Network commands, configuration files or API calls to administratively enable a switch port and assign appropriate VLANs; add a server to a load balancer; add a firewall rule; or check if a port is online. This also includes the relevant rollback configuration associated, in case it is needed.</td>
</tr>
<tr>
<td>Verify</td>
<td>Establish the proposed monitoring and operational activities to verify specific network activities or changes go as planned. The output is likely a script or API call.</td>
<td>Network commands to verify a switch port is online and the correct server is attached, and that the port is passing traffic not taking errors. This may also include verifying the syntax validity of a configuration via a vendor’s management system or Batfish.</td>
</tr>
<tr>
<td>Preproduction</td>
<td>Test the proposed network activity and associated verification tasks prior to deploying to the production environment. This could be done in a lab or virtual environment.</td>
<td>Use a vendor’s virtual environment or GNS3 to validate that a proposed change to a network device works as expected. There are also tools such as Forward Networks and Batfish that can be used for this as well.</td>
</tr>
</tbody>
</table>
As organizations mature, each of the network activities above should be exposed via APIs so it can be called programmatically (see Note 3 and “How to Automate Your Network Using DevOps Practices and Infrastructure as Code”).

What About Change Control?

Many networking teams report concerns about network change control policies. Rigid networking change control and DevOps do not mix well. Thus, change control policies must be adjusted to allow for network changes to occur more quickly as part of a DevOps initiative (see “How to Establish an I&O Change Management Process That Supports Agile Development and DevOps”).

As more projects are delivered using DevOps practices and technology, I&O leaders must enable smaller, more frequent change windows (instead of weekly or monthly “big bang” maintenance). For example, I&O leaders can designate a specific time of day (such as 5 p.m.) to implement a subset of approved changes, such as enabling ports or adding servers to load balancers, which are lower risk (see “Use Adaptive Release Governance to Remove Constraints to DevOps”).

Evidence

1 dgarros/netdevops-survey. This open, public online survey was led by Damien Garros (Network to Code) and Francois Caen (Cisco) and was completed by 293 participants from 10 October 2019 through 10 November 2019. Gartner assessed this survey as being largely vendor-agnostic and representative of enterprises that are more advanced in their network automation journeys.

2 Gartner client inquiry, which includes:

- More than 5,000 calls regarding networking from February 2019 through February 2020.
- More than 5,000 calls regarding DevOps from February 2019 through February 2020.
Gartner 2019 DevOps Survey. Results presented are based on a Gartner study conducted to assess the objectives, performance and challenges faced in DevOps initiatives. It also delves into the performance, drivers and challenges of scaling DevOps. The primary research was conducted online from 14 November 2018 through 18 December 2018 among 273 respondents in North America, Western Europe and Asia/Pacific. Qualifying organizations span various industries, except “services.” Companies were screened for having annual revenue for fiscal year 2017 to be greater than/equal to $100 million and to have a minimum 50 full-time IT employees. Companies were required to have DevOps to support systems/IT products in production. DevOps efforts were required to be completely in-house or a mix of in-house and outsourced with a minimum five DevOps teams to support systems/IT products in production. The sample represents organizations in the U.S. (n = 83), Canada (n = 35), the U.K. (n = 44), Germany (n = 31), India (n = 48) and Australia/New Zealand (n = 32). Respondents were required to have a role that is primarily IT-focused or be a fairly even blend of business and IT. They were also required to have involvement in decisions regarding DevOps efforts at their organization. Quotas were applied for countries, industries and annual revenue. The study was developed collaboratively by Gartner analysts and the Primary Research Team. Disclaimer: Results do not represent “global” findings or the market as a whole but reflect sentiment of the respondents and companies surveyed.

Note 1. Popular Tools for Automating Networks and Applying DevOps Principles to Networking

Representative vendor solutions include:

- **Ansible** — Open-source network automation software that is agentless and supports multiple devices. Commercial support is provided by Red Hat. Ansible is widely supported by nearly all commercial networking vendors who provide modules that help automate specific networking activities.

- **Batfish** — Open-source network validation software that can help perform prechange validation. Commercial support is provided by Intentionet.

- **Git** — Open-source software version control repository which can be used for network device configuration files, and templates. Github is a cloud-based service for managing Git, and offers a freemium tier.

- **GNS3** — Open-source network simulation software platform that allows for testing of network device configurations including Arista Networks, Cisco, Cumulus Networks, Extreme and others.

- **Jinja2** — Open-source dynamic templating framework for Python that is used by Ansible.

- **NAPALM** — Open-source, Python-based library that allows for automation of network devices. It can be used as a multivendor API.

- **NetBox** — Open-source IP address management (IPAM) software that is used to help build a source of truth.
Note 2. Site Reliability Engineers and Automation Architects

Site reliability engineers combine software development, networking and system engineering expertise to build and run fault-tolerant software systems and infrastructure (see “Hiring Guide for Site Reliability Engineer”).

Automation architects assume primary responsibility for the organization's automation program, including the technical work and decision making involved in building the automation platform (see “Leverage an Automation Architect Role to Accelerate I&O Automation”).

Note 3. Enabling Network Activities

Ideally, network activities would be driven by integrated tools in the development environment (such as Jenkins, Chef, Puppet or Ansible) to avoid making developers leave their native toolchain environment. However, in practice, network activities will almost certainly require network-specific automation platforms. Representative vendors include Cisco (Prime), Micro Focus and SolarWinds (for more vendors, see “Market Guide for Network Automation”).

Document Revision History

Best Networking Practices in a DevOps World - 25 September 2018

Recommended by the Authors

How to Reduce Technical Debt in Enterprise Networks
Cool Vendors in Enterprise Networking
DevOps Teams Must Use Site Reliability Engineering to Maximize Customer Value
Balance Velocity and Risk by Having DevOps Teams Earn Leaner Processes
How to Build and Evolve Your DevOps Toolchains

Puppet — Open-source automation software widely used in the server/compute arena, but also extends to networking devices with support from many networking vendors. Commercial support is provided by Puppet Enterprises.

Python — Open-source programming language with clear syntax and improved readability compared to other programming languages. Python is popular for network automation activities.

Nornir — Open-source automation framework that is intended to be run directly from Python.

SaltStack — Open-source automation solution.

TextFSM — Open-source Python module that can be used to parse device output (such as device CLI commands) into formatted structure or table.