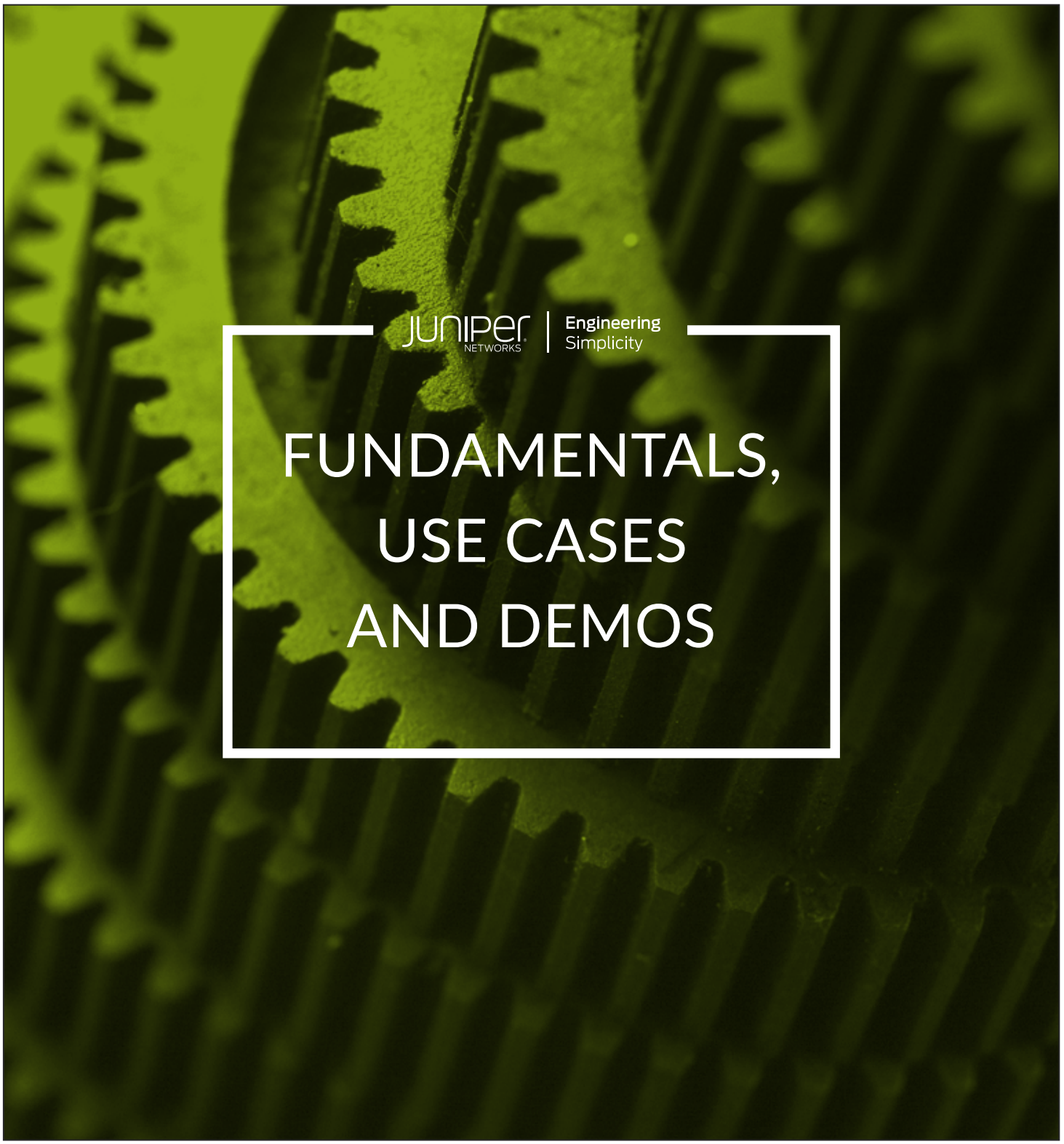
ENGINEERING AUTOMATION

WEBINAR MINISERIES



JUNIPER.NET/AUTOMATE

INTRODUCTION

In a world shaped by technology, automation helps businesses move fast and pivot on a dime. Meanwhile, the foundational value of network operations is steadfast reliability, a condition that is traditionally best during periods of inactivity. "Network Reliability Engineering" (NRE) is an emerging approach to network automation that stabilizes and improves reliability while achieving the benefits of speed.

Implementing DevOps Transformation as Reliability Engineering

DevOps defines speed as providing faster feedback and reducing lead times through deliveries with a small-batch cadence. Indeed, this is a proven recipe for continuous improvement, and one that requires automation coupled with an automated pipeline to drive continuous response improvements in system regulation, capacity planning, and feature planning.

DevOps also introduces many cultural principles and practices. In situations where DevOps does not provide job descriptions or implement abstract processes with substantive technology, however, site reliability engineering (SRE) steps in, offering a solution for software delivery and operations teams. As SRE grows in popularity, network operations teams looking to redirect their approach to network automation have assigned a similar moniker to their own processes—network reliability engineering (NRE)—and refer to their people as "network reliability engineers."

Beyond applying SRE behaviors and tools to network operations, it's easy to appreciate why network automators prefer to identify as engineers rather than developers and why they value reliability over speed.

Approaching Automation as Engineering

While a significant amount of automation can take place on the vendor engineering side, last-mile contextual work is always required to glue together the various customer-specific systems integrated with the network. Every network team has some unique workflows that allow humans to interface with networking systems in order to make changes to provisioned intent and view the operational state.

Approaching these tasks with human technicians is an IT antipattern that should be avoided at all costs. In order to improve human accuracy and operational simplicity, the contextual automation engineering that takes place inside network operations must pick up where the vendors leave off.

Forrester recently reported[1] that mainstream enterprises are applying SRE to their IT infrastructure and operations. While NRE is just getting started, one of the positive aspects of this trend for a network automation industry that has focused almost exclusively on technology at the expense of processes is that it applies the rigors and proven processes of software engineering to the automation journey. For network engineers, learning and applying software engineering skills and processes to network operations is different than application developers doing software engineering by vocation.

For most network engineers, their introduction to automation was automating and aggregating traditionally manual operations. Before they could engineer their workflows, they first had to recognize and identify them. Going beyond tribal knowledge and limited documentation is often the first step toward automating routine and repetitive tasks, introducing consistency, accuracy, and reliability. Progressing in this way, over time, more work can be automated.

This type of advancement in workflow automation contributes to learning and productivity, but it does not necessarily transform how people look at network operations. Rethinking operations as a rigorous exercise in software engineering, network reliability engineers consider building, testing, staging, and stressing the boundaries of their network architecture and automation technology. This requires aligning with proven processes like gitOps, automation, and testing with source-code management, infrastructure configurations as code, code reviewing, preproduction pipeline orchestration with continuous integration and delivery (CICD), and other patterns explored in the 5-Step Journey to Automated NetOps.

1 https://reprints.forrester.com/#/assets/2/157/RES142216/reports

CAST OF CHARACTERS

JAMES KELLY
MARKETING DIRECTOR AND SOFTWARE ENGINEER, JUNIPER NETWORKS

DERICK WINKWORTH
SENIOR PRODUCT MARKETING MANAGER AND ENGINEER, JUNIPER NETWORKS

DWANN HALL
SENIOR PRODUCT MARKETING MANAGER, JUNIPER NETWORKS

MATT OSWALT
SENIOR PRODUCT MARKETING MANAGER AND ENGINEER, JUNIPER NETWORKS

DAVID GEE
AUTOMATION SPECIALIST SYSTEMS ENGINEER, JUNIPER NETWORKS

CHRISTIAN GILBY
PRODUCT MARKETING DIRECTOR, JUNIPER NETWORKS

MATT BARLETTA
DIRECTOR OF TECHNICAL MARKETING, JUNIPER NETWORKS

TABLE OF CONTENTS

EPISODE 1

GETTING FROM AUTOMATABLE TO AUTOMATED

Engineering Automation Series

WATCH THIS EPISODE

CAST

James Kelly

— Marketing Director and Software Engineer, Juniper Networks

James loves helping businesses transform with technology. Using his skills with cloud-native design, DevOps, SDN and infrastructure as code, James's software and business experience have been applied over the last decade at Juniper Networks in the areas of customer-facing engineering, business development, product management, and he's now Marketing Director of Enterprise Networking and Automation. Prior to working at Juniper, James was a tech researcher, developer, hedge-fund founder, and executive technology consultant.

Episode Outline

- IT automation: Status quo isn't static
- State of network automation
- What does success look like?
- A 5-step journey
- Juniper's help for your journey

Episode Insights

More than a decade of focus on network programmability, SDN, NFV, APIs, and other tools have made network automation all about the technology. However, "automation" hasn't necessarily led to "automated." Today's challenges lie in the consumption of the tools and technologies available to network engineers. Process transformation and training will be the defining factors in moving networks from automatable to automated, putting programmability into action

When it comes to automation, ironically, humans are the heroes

Vendors can make the network easier to orchestrate and automate, but they cannot automate network operations.

In conversation about automation, it's tempting to put technology front and center. And it will certainly play a role in software-defined, intent-based, and more autonomous networks. However, in the end, network engineers are the heroes when it comes to consuming and delivering automated network operations.

Narratives that paint a hapless picture of the future by eliminating human operators have it backward. In fact, SRE and NRE are often described as roles ("engineers" instead of "engineering"), placing people at the center of change, giving them more attention, more responsibility, and greater rewards.

Before long, network engineers will be more technologist than technician. They'll barely touch device CLIs, but it won't be all about GUIs, either—they'll shift to APIs that drive higher-order workflows and gitOps changes. No matter the provisioning altitude of intent and the amount of logic, actuators, and autonomous sensors in a system, humans will remain the key interface that drives change and gathers information about the system state in order to make decisions and manage network services.

Automation technology will reduce the daily toil of repetitive tasks that leads to unintended mistakes. It also provides the guardrails that ensure adherence to SLAs. Network SLAs and reliability are not left to caffeine-powered individual heroics, but are achieved through well-trained automation heroes known as network reliability engineers and the power of NRE.

Episode Q&A

How do you explain that automation is not a product?

As we went through a worldview of network automation, we took a closer look at NetOps and the network itself, focusing on the operations side. We examined some of the products, APIs, and tools needed, but found that the crux of automation is engineering operations such as troubleshooting, change workflows, test validation, event sensors, processing and reactions, system integrations, and building stakeholder service-level indicators.

How does chaos engineering relate to SRE and NRE?

Continual improvement requires us to learn, and learning requires us to allow for failure. Chaos engineering is a concept and methodical practice that causes reproducible failure experiments. These allow us to embrace failure in a way that allows us to observe, learn, and then automate around those failures, improving reliability.

The 5-step process suggests early automators start with troubleshooting workflows. Yet today, many start with configuration management. Why do you recommend this approach?

Most network engineers spend more time troubleshooting and understanding the state of the network and less time making intent or configuration changes. That means there is a greater benefit to automating tasks that are performed more often, but even if that were not true, configuration workflows are inherently riskier because they make changes. Read-only tasks like troubleshooting are a safe place to start in production, especially for many network engineers who may not have a development, testing, and staging lab.

EPISODE 2

NETWORK AUTOMATION IS ABOUT RELIABILITY

Engineering Automation Series

WATCH THIS EPISODE

CAST

DERICK WINKWORTH

— Senior Product Marketing Manager and Engineer, Juniper Networks

Derick has been a software and network engineer throughout his career, working in many different industry verticals. He enjoys playing video games with his kids, grilling in the snow, and has a 250+ t-shirt collection. He works on Juniper's Enterprise Marketing team as the developer advocate for NRELabs, a Juniper sponsored, multivendor, open-source tool for learning about automation.

Episode Outline

- Networks are fragile
- Organizational trust
- Automation and reliability
- A path forward

Episode Insights

For many people, the allure of automation is speed. In this webinar episode, we dispel the myth that speed is the ultimate benefit of automation, and look at why this myth—and others—not only exist, but persist.

IT workers often believe that businesses must sacrifice speed to realize other gains like efficiency and reliability. To keep up with the demands of the business, speed and agility are top of mind. But focusing on speed alone is actually a threat to reliability. Network engineers must prioritize reliability first or risk tarnishing organizational trust that empowers them to make changes.

Automating reliability first and speed second

Even in F1 racing, a sport defined by speed, there is a saying: it's not how fast you drive, it's how you drive fast, and in order to finish first, you must first finish.

One important difference between delivering software for business services rather than for network infrastructure services is that networks do not demand as much continual innovation and tweaking. Therefore, in infrastructure—especially the network—reliability is intuitively more important than moving fast. Even for those looking to move faster to keep up with DevOps-era software engineering demands for on-demand elastic infrastructure, without reliability, speed is irrelevant. Interestingly, however, when asked to think of words to describe their network, "reliable" is almost never mentioned by network engineers. This episode explores ways to change that, setting upon a journey to automate with NRE thinking and practices.

Episode Q&A

What is the book on SRE that parallels the NRE topic discussed in this episode?

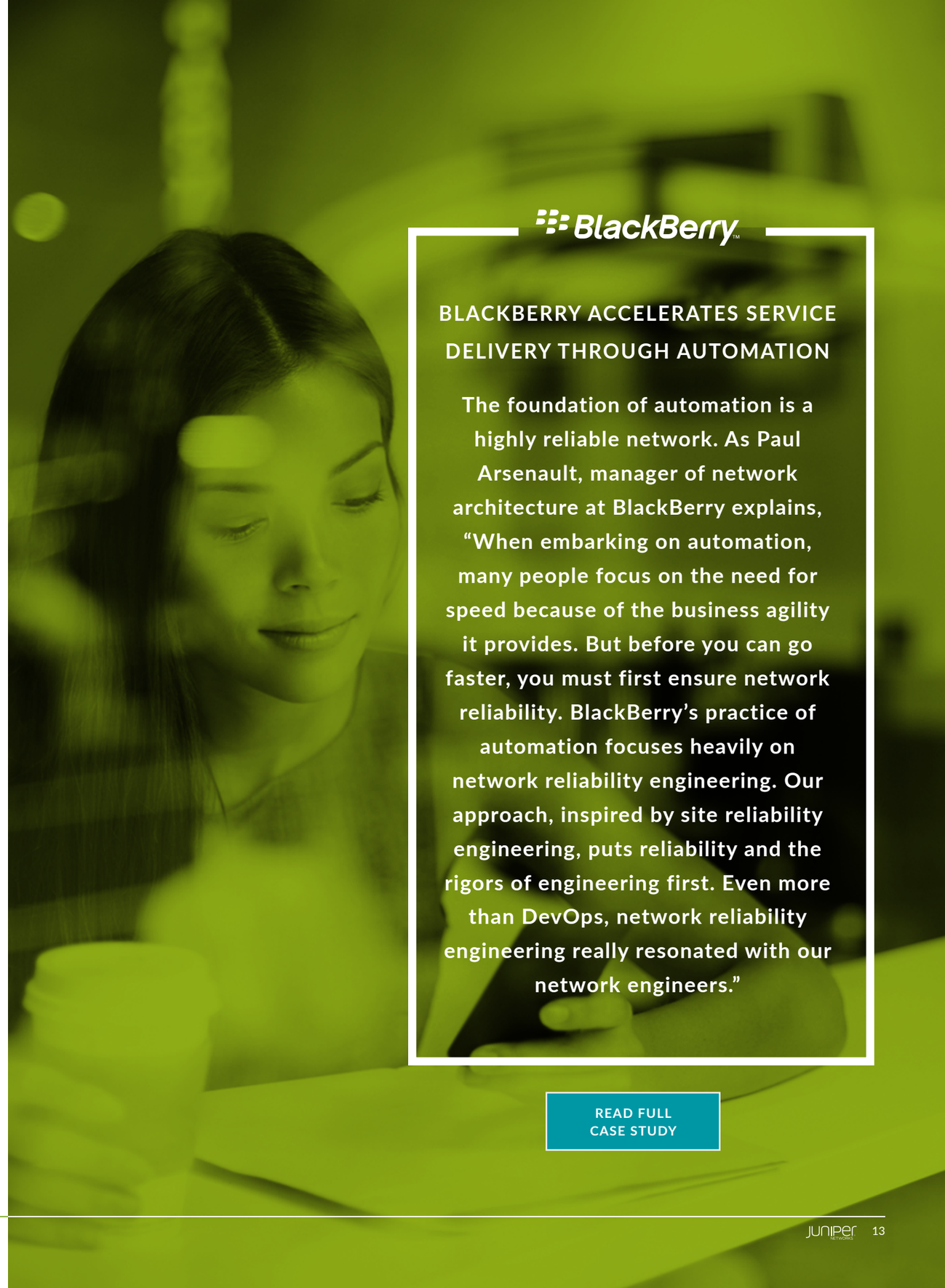
Site Reliability Engineering, or SRE, is the name of a practice and a book published by Google. SRE has become a popular implementation of DevOps, with prescriptive practices in automating for reliability first and foremost. Network Reliability Engineers and Engineering (NRE) was introduced by network engineers later, obviously inspired by SRE. Many SRE practices translate to NRE. The SRE book, published by O'Reilly, is available online for free here.

What tools are used in network automation?

There are too many tools to list, and they vary by type and purpose. Some are proprietary and productized. Others are open source. Some examples include Nornir, Ansible, Netbox, NAPALM, and a few related to Junos OS such as PyEz and JSNAPy. While we've already discussed the NRE Labs website, it's worth mentioning that the NRE Labs community has a social site and resources, including a poster which classifies some tools as something that parallels the 5-step framework shown in this webinar episode.

What do you say to folks who believe automation is about reducing headcount?

While we addressed this as one of the myths in the webinar episode, this is a real concern among IT practitioners in general, not just network engineers. Maintaining an automation practice, including development, testing, and staging, requires additional work over and above network architecture and change management. It's true that this will reduce rote change workflows, but the time saved will be spent in other areas of creative engineering.



EPISODE 3

VENT-DRIVEN SECURITY: AUTOMATED DETECTION AND ENFORCEMENT

Engineering Automation Series

WATCH THIS EPISODE

CAST

DWANN HALL

— Senior Product Marketing Manager, Juniper Networks

Dwann has a passion for helping customers better serve their own customers and helping them achieve desired business outcomes. He has a unique talent for articulating complex solutions while relating them to the business problems that a customer is experiencing, or growth opportunities that they may be exploring.

Prior to joining Juniper in 2012, Dwann served in a number of Architect and Sr. Engineering roles building out Internet scale data centers, serving users in the hundreds of millions.

Episode Outline

- Theory of constraints
- Incident response frameworks
- Dwell time and security automation
- Example: SaltStack
- Example: Juniper connected security

Episode Insights

When defenses have been breached and security has been compromised, the clock starts ticking on the metric no one in the industry wants to talk about: dwell time. Dwell time is the gap between when an incident occurs and when it has been detected and contained. Recent estimates show dwell time increasing year over year, currently sitting at around 100 days—and that's just the average!

Decreasing dwell time with automation

To combat advanced threats and reduce dwell time, much of the focus of security automation has been trained on faster detection. A number of products and solutions that employ machine learning and artificial intelligence centered around behavioral analytics and event correlation to detect threats faster are available. While this is effective in today's reality of alert fatigue, too much data, and not enough people, incident response is still a workflow.

Advances in security information and event management (SIEM) and security orchestration, automation, and response (SOAR) tools now provide the mechanism to trigger a response. More often than not, that response is to open a ticket for the security operations center (SOC) or to assign a human to perform manual investigation, judgment, and containment. As detection time falls, containment time becomes the bottleneck.

In this webinar episode, we look at how event-driven frameworks like SaltStack and Juniper Connected Security's Policy Enforcer use automation to reduce containment and dwell time.

Episode Q&A

How is Juniper Connected Security different than my ATP and endpoint security solution?

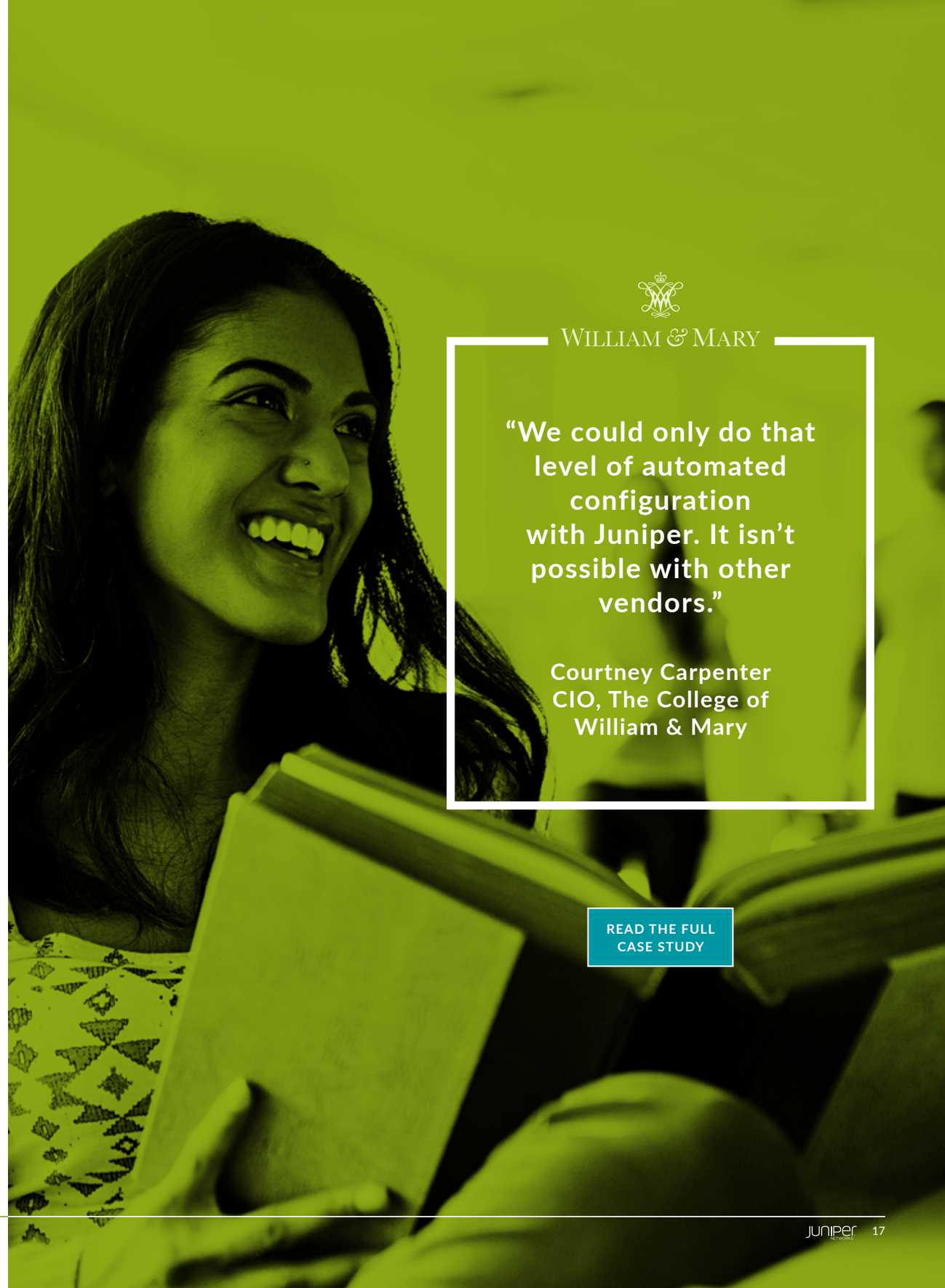
Endpoint security solutions are an important part of any security posture, and Juniper partners with several leaders in the endpoint market. While they're a necessary component to protecting your enterprise's managed hosts, they're not sufficient in the age of IoT and BYOD where it's not feasible to secure these unmanaged devices with endpoint security. Juniper Connected Security, as shown in the webinar episode, enables device blocking and quarantining in the access networks, even without an endpoint solution.

I'm worried about accidentally blocking my executives and users. How do false positives factor into automating enforcement?

We remember the early days of prevention and access control systems where good traffic was falsely identified as a threat. Remember that event-driven security essentially revolves around detection and enforcement, automating enforcement policy actions and updates. You can keep your existing detection workflow to identify which targets you want to block, but once that decision is made, you can automate the enforcement action.

How does event-driven security compare to SIEM and SOAR?

Event-driven security is complementary to security information, event management (SIEM), and security orchestration, automation and response (SOAR) tools. A SOAR is about automating processes, one of which is to block or quarantine a compromised host. Event-driven security and Juniper Connected Security can automate that workflow. A SIEM will collect and correlate data to find threats and provide observability. Many of them generate offense events that need to be acted upon. These triggers can be handled with event-driven security frameworks, taking an action such as running a script or updating a policy that would result in automatic enforcement.



EPISODE 4

TEST-DRIVEN NETWORK AUTOMATION

Engineering Automation Series

WATCH THIS EPISODE

CAST

MATT OSWALT

— Senior Product Marketing Manager and Engineer, Juniper Networks

Matt hails from Portland, OR, and focuses on the intersection of network infrastructure, automation, systems, and software engineering. He's passionate about enabling engineers to evolve their careers to the next level, and sharing the bright spots that exist within the technology industry with the masses.

You can often find him speaking at conferences or meetups about these topics, as well as writing about them on his blog https://keepingitclassless.net or on Twitter as @Mierdin .

Episode Outline

- Why testing is important
- Test-driven network automation
- Demo
- How to get there

Episode Insights

While testing is central to boosting network reliability, it is often an afterthought. It doesn't have to be. NetOps teams can borrow from test-driven development to put testing front and center. Even without fully test-driven development practices, adding some testing to automation workflows reduces and expedites troubleshooting. Many reactive troubleshooting automation tools can also be repurposed as proactive tests.

Slow and Steady Wins the Race

For both software and network engineers, testing seems like a burden that will slow down operational processes. In reality, while automating testing for compliance, security, functionality, and even non-functional service levels may be time consuming upfront, it eliminates a lot of later rework, regressions, and repetitive troubleshooting tasks. As discussed in prior episodes, reliability provides speed and agility as incidental benefits.

What You Know That Just Ain't So

You may believe that what you don't know can hurt you. In reality, it's what you know that just ain't so that can really come back to bite you. As explored in this episode, network engineers' idea of what it should be (WISB) often doesn't match what it really is (WISI), and these gaps are where reliability often falls flat. That needn't be the case.

The crux of testing is codifying tribal knowledge and enforcing validation of those intentions.

This episode examines various ways to build and automate testing for network operations, showing demos of testing configurations as well as the network's running state. It also explores more integrated forms of active and passive testing on traffic itself.

Episode Q&A

Where does one deploy testing?

Tests can run in a variety of places—for instance, in the production network on devices and hosts. Moreover, testing or virtual labs like Juniper's Cloud CCL are good staging environments where you can safely replicate a production setup and test changes in an integrated fashion before committing those changes to production. Looking at Step 4 in the Automated NetOps 5-step Framework, we see that continuous integration (CI) is an evolution of ad hoc testing in Steps 2 and 3. Starting at Step 3 with infrastructure as code, tests can run at commit time with Git hooks and, in Step 4, through the entire build-test CI pipeline. Pipeline tools like Jenkins are often used to orchestrate this preproduction process.

What kinds of tools are used for testing?

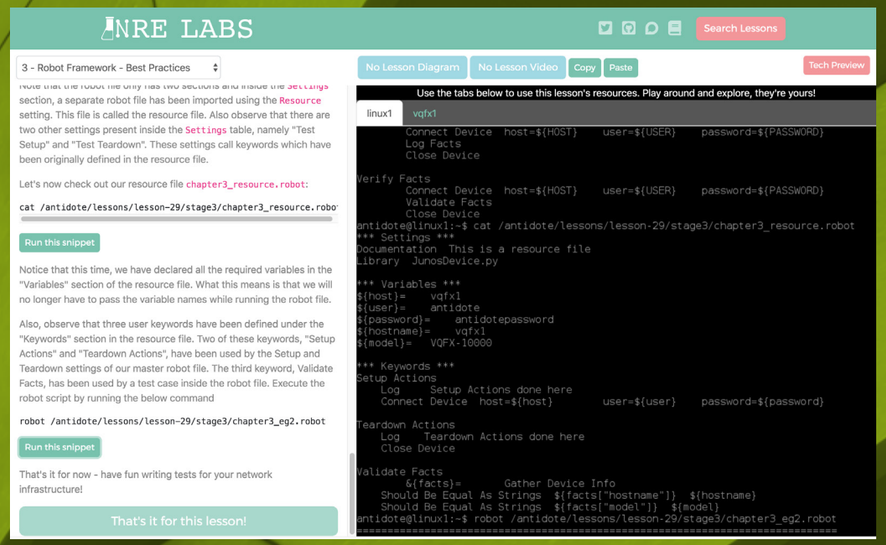
This is by no means an exhaustive list, but JSNAPy for Junos, NAPALM, Python for hand-written tests, Behave, Robot Framework, and Batfish are all examples. The important thing is that people do not get stuck trying to analyze the whole testing space and all its tools before they start on the testing journey.

Where can I go to access the testing lessons demoed in NRE Labs?

NRE Labs is sponsored by Juniper, but freely open to anyone to use and contribute to. All you need to do is search for NRE Labs. The lesson shown in this webnar is on JSNAPy. The community home page has more information on NRE and NRE Labs.

NRE LABS

An automation dojo in your browser. Learning to automate is now free, open, easy, and fun.



LEARN MORE

EPISODE 5

CODIFYING NETWORK CONFIGURATION THROUGH SIMPLE DECLARATIVE

Engineering Automation Series

WATCH THIS EPISODE

CAST

DAVID GEE

— Automation Specialist Systems Engineer, Juniper Networks

Despite learning to weld recently and being quite excited about it, David dedicates his professional life to building control and instrumentation systems focussed on IT infrastructure and specifically network automation. He brings together extensive knowledge on industrial automation, embedded systems and real-time mission critical real-time software design to make network operations reliable through modern automation technologies. David also has history of working with emerging technologies and tracked SDN from the first industry whispers along with system programmability. He works for Juniper in EMEA as an Applied Automation consultant.

Episode Outline

- State of network automation
- We can do better
- Declarative approach
- Where to go

Episode Insights

"Simplicity is prerequisite to reliability," said Edsger Dijkstra, famed computer scientist and network algorithms architect. In the quest to automate reliability, this episode examines how a declarative approach to provisioning intent removes complexity, swaps tribal knowledge for codified clarity, and abstracts the interface to manage state safely with a modern DevOps paradigm implemented with Terraform tooling.

Declarative Infrastructure as Code

Network engineers are classically CLI experts on network devices. But in a collaborative culture, pockets of expertise can be bottlenecks. Declarative configuration is all about the "what," not the "how." It's a move toward simplicity and a better approach to operations and automation.

Elevating the level of abstraction for provisioning above the minutiae and intricacies of network primitives and syntax, Terraform provides a way to declare what you want, and then let the tool perform the creates, reads, updates, and deletes quickly, consistently and reliably. Moreover, it's a perfect match for GitOps and testing discussed in other episodes.

Demo: Seeing is Simplicity in Action

A demo shows the power of declarative simplicity in action. Take the familiar example of VLAN provisioning: it requires a VLAN, an access port, and a Layer 3 interface for routing. In just a few minutes, David explains the short Terraform configuration files and Terraform lifecycle workflow required to plan, graph, apply, refresh, and, finally, destroy network infrastructure. Infrastructure as code isn't just for cloud resources. Terraform, a leading tool for multicloud infrastructure automation, now unleashes declarative automation for networks.

Episode Q&A

What Terraform providers are supported, and are they supported by Hashicorp?

The module for Juniper Networks QFX Series data center switches comes first. That is what was demonstrated and is currently available to experiment with in NRE Labs. The release is coming soon, followed by planned support for features specific to routing on the MX Series routers and security on the SRX Series firewalls. Juniper has also open-sourced some Terraform providers for Contrail products.

The Terraform providers for Junos will be merged upstream in the Hashicorp codebase and main Terraform release, so that Terraform will come with them. However, Juniper will be responsible for supporting their functionality. Community Q&A support is available in the Slack space on EngNet as well.

Is Terraform open source?

There is a popular free community edition called Terraform Open Source as well as a commercial editions by Hashicorp called Terraform Cloud and Terraform Enterprise (TFE) that make it easier to use Terraform across teams and large organizations.

Are the .tf files used stored in a Git repository?

They can be. It's good practice to store infrastructure as code configurations, scripts, and source code in a system like GitHub or GitLab to version control all the files and spur collaboration. As part of automating your pre-production processes, along with binary artifact repositories, these systems allow for smooth integration with CI/CD pipelining tools for testing, building, integration and deployment.

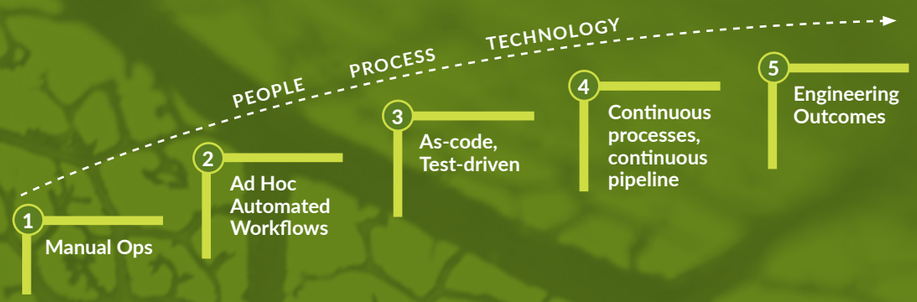
THE AUTOMATION JOURNEY

The challenge lies not only in knowing where to go, but how to get there.

Automation has become imperative to modern network operations. You need it within the products you use to build your network to make it more autonomous. It's also critical to enabling reliability in your network operations processes. But not everyone knows how to get started with automation, how to set long- and short-term goals for achieving it, and how to measure success.

Getting there certainly raises technical challenges that organizations must address. Equally important, however, are changes in processes, skill sets, and culture. All three areas—people, process and technology—must evolve in parallel to accomplish the ultimate automation goal, a more reliable network infrastructure, and such secondary goals as speed, efficiency, and agility.

The approach to automation as a network reliability engineering journey can be summarized in five steps:



Use the many resources on this page to learn how Juniper can help you successfully follow this path to achieving more reliable network operations.

WATCH VIDEO

EPISODE 6

AUTOMATING YOUR AI FOR IT™ NETWORK WITH MIST

Engineering Automation Series

WATCH THIS EPISODE

CAST

CHRISTIAN GILBY

— Product Marketing Director, Juniper Networks

Christian has 20+ years of product marketing, management and engineering experience in the networking industry with a strong focus on mobility, cloud and wireless and speaks often at industry events. He leads product marketing for Mist (acquired by Juniper). Previously he led product marketing for wired, wireless and branch solutions at Aruba (acquired by HPE), and previously at Agito Networks (acquired by ShoreTel), Meru Networks (acquired by Fortinet) and Nortel.

MATT BARLETTA

— Director of Technical Marketing, Juniper Networks

Matt is a passionate technologist with 20+ years of network architecture and wireless expertise. As the Director of Technical Marketing, Matt has been hands-on with Mist from day 1 and has been involved with many of Mist customers. He believes strongly in leveraged content to share these experiences with as many people as possible through web, courses and webinars. Previously he led teams at Cisco, Airespace and Violin Memory.

Episode Outline

- The Mist journey
- APIs and AI
- What are the APIs and why to use them
- Tour of the Mist API
- Let's get programming
- Learning resources

Episode Insights

While the ability to automate networks with APIs is almost ubiquitous among network vendors today, Mist is different. Mist through out the old model of building monolithic and shrink-wrapped software thanks to its foray into business in the era of cloud native. Mist was born in the cloud with a microservices architecture, providing feature agility, but it surpasses other systems when it comes to programmability.

So often the API is an afterthought, and vendors are creating APIs from features in GUIs and CLIs. But in the Mist architecture, every microservice communicates with every other microservice through APIs; thus every piece of information is created, read, updated, or deleted through a REST API. All those APIs are at the network engineer's disposal to automate their operations. Imagine that some of Mist's customers operate 100% through the Mist API!

From API Novice to 500 Sites in Mere Minutes

In this episode, we show numerous examples of how to use the Mist APIs. For those who have never touched an API or done any programming, Matt explains how REST APIs are actually just HTTP calls.

You can get started with free web-based tools like Postman to make API calls without having to write a single line of code.

From there you will jump into the helloworld of Wi-Fi, showing you how, with just a few lines of Python, Matt can list the networks on his wireless network. With just a few small additions, we can generate and apply a new password on a network and automatically send an email to the administrator with the new password. This has real-world applications when you consider a guest signing in at reception and sending them an auto-generated password.

Want to see how many iPhones are running the latest version of iOS? You'll see how simple it is with Mist, and how you can string your scripts together with common Linux command-line tools like grep. Finally, we'll show a small 184-line Python program that digests a spreadsheet of 500 site names and addresses and, using a Mist template, creates all sites in just minutes and with zero errors. This webinar is sure to spark countless automated NetOps workflows. Thankfully, you have a running start with so many resources available on Mist's GitHub site that will get you automating in no time.

Episode Q&A

What was the environment you used for the Python coding? What labs are in the Mist API course?

https://repl.it is indeed free, but we also ran Python scripts right from a Mac laptop. The Mist course materials are available at https://api-class.mist.com/. In this course, you get hands-on access, learn what APIs are available, and learn tips and tricks for automating with Mist. It's all online in a pre-configured sandbox environment. You can also find examples at https://github.com/mistsys/.

Can anyone view the Mist APIs?

Yes, if you have a Mist account or create one, you don't have to be a customer to view the APIs. We are looking at incorporating this with Juniper EngNet. API documentation is available at https://www.mist.com/documentation/category/API.

Are there rate limits on API calls?

We limit API calls to 5,000/hour, although for some large customers, we have expanded that. We usually want to know what they are doing and sometimes we can suggest more efficient ways to solve their problem.

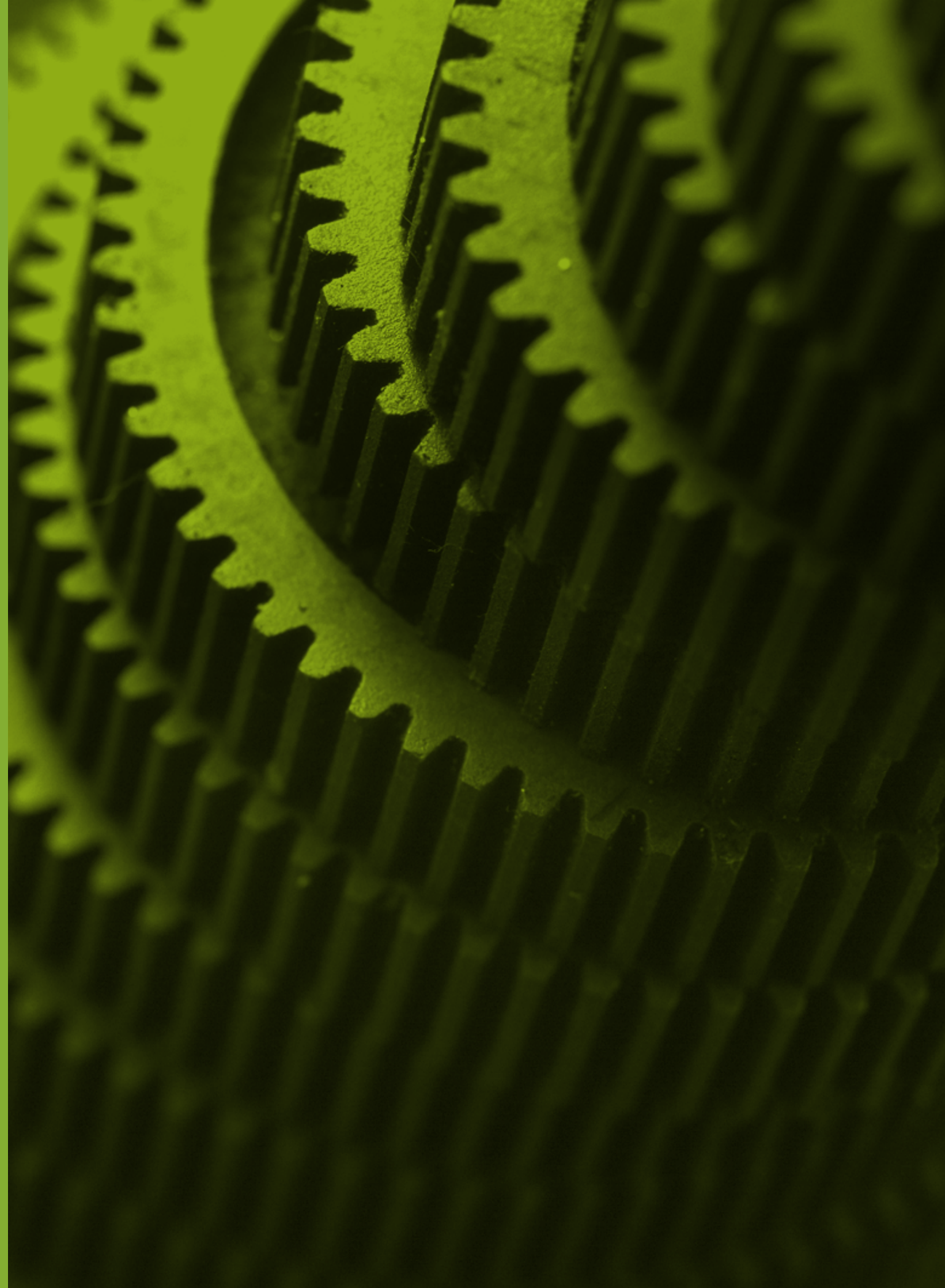
Is the Postman API testing tool free?

For what we are showing, yes. You only get charged when you need to do more advanced automation, like for automated testing.

JUNIPER NETWORKS | Engineering Simplicity

ABOUT JUNIPER NETWORKS

Juniper Networks simplifies the complexities of networking with products, solutions and services in the cloud era to transform the way we connect, work and live. We remove the traditional constraints of networking to enable our customers and partners to deliver automated, scalable and secure networks that connect the world.



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