

**FREE
DVD**

Apache Mahout

Defense by Design

ADMIN
Network & Security



ADMIN

Network & Security

ISSUE 70

Defense by Design

- Extended detection and response
- Ransomware contingency plans
- DNS cyber defense

APACHE MAHOUT

Distributed linear algebra framework

Azure AD with Conditional Access

Jaeger

Troubleshoot cloud-native applications

Prometheus Anomaly Detector

AD Sites and Services

Replication between sites

Puppet Bolt

Automate admin tasks

Foundries.io

Develop IoT apps for devices

Teleport

Centrally manage logins

Weka

Statistics and ML

LINUX NEW MEDIA
The Pulse of Open Source

Get started with



SysAdmin JOB HUB

Top jobs for IT professionals
who keep the world's
systems running

SysAdminJobHub.com

The Great Stay Put

The IT field is either feast or famine, depending on your perspective, but one thing is certain: No matter where you decide to land, you always have value.

A few months ago, I wrote this column describing “The Great Resignation.” Now, I think I must write this one as “The Great Stay Put,” because things have changed so rapidly since then. I had real wide-eyed optimism about the IT job market, but all that has changed. Now, just as was seen in the early 2000s, you might be lucky to have a job in six months. I hope that’s not the case, but you need to prepare yourself. One way to prepare is to change to a safer position, and by “safer,” I mean one that is essential to business operations. The other way is to stay where you are and ride this thing out. There’s no way to predict accurately what’s going to happen six months from now, but that doesn’t mean you have to sit around and just wait for it to happen to you. You can act now to be sure that you decrease any negative effect from a downturn in the economy.

Whether you decide to stay or leave your current job, you must position yourself to show that your presence is important to the ongoing health of the business. The best way to do this is to examine the business, its strengths, its weaknesses, its assets, and its value to customers. You must find ways to improve production, cut costs, increase revenue, and resonate better with your customer base. What are you doing right? What are you doing wrong? Are you leaving money on the table by not cross-selling, upselling, or enhancing the customer experience with better service and new or upgraded products?

You must find some way to contribute in a more meaningful way to help solidify your stake in the company’s success. I guess that’s a fancy way of saying that you need to participate actively in increasing your company’s bottom line. Be aware that not every idea is going to be great. It’s likely that your management will reject most of your ideas. Don’t allow rejection to discourage you. Brainstorming is how people discover great things, so keep at it. Also, remember that everything works on paper. I’ve had some incredible ideas that worked perfectly well on paper, but two minutes in front of someone else’s eyes yielded a disappointing, “Yeah, but what about X?” Too often my response was, “I hadn’t thought of that. I guess I need to examine this more carefully.” Thomas Edison didn’t create the light bulb on his first try, and the Wright brothers flew more than 1,000 glider test flights before putting engines on their first powered airplane. One of the things you must do is write down your ideas as you think of them. Write each new idea at the top of a page. This way you can expand on your idea, list the pros and cons, and write out how you can implement or deploy your plan as you develop it. Don’t be afraid to list crazy things in your notebook. No one else ever needs to see it. Present your best ideas to your manager, showing that you have considered the idea from different perspectives. Create a presentation along with your idea to show that you have presentation skills and that you’re really putting some effort into these new ideas, whose goal is to help the company. You and your manager will both appreciate your ideas and analyses, even if they’re rejected.

As a real-world example, I had a former co-worker who went through this exact process, made his presentation, and not only gained acceptance for his idea but was promoted to my manager’s manager in the process, where he implemented his plan. I give him props for doing this, although it ultimately led to my leaving the company because of our difference of opinion on our team’s goals and direction. However, the experience taught me something valuable: Present your ideas in an organized, determined manner and see what happens. Sometimes it works out.

I hope that if you decide to stay with your current company, you take some of the advice I’ve given to help you remain gainfully employed. You’ll no doubt see layoffs and voluntary attrition in the process, but in the end, sometimes staying the course where you are is the right thing to do.

Ken Hess • ADMIN Senior Editor

ADMIN

Network & Security

Features

Nothing is so true in IT as “Prevention is better than the cure.” We look at three ways to prepare for battle.

10 XDR

Extended detection and response integrates security functions across endpoint devices and networks. But is XDR the only integrated approach to cybersecurity challenges? We investigate the new technology.



16 Defense Against Ransomware

The possibility of a ransomware attack means it is essential to prepare for cyberattacks by putting defense mechanisms and contingency plans in place.



20 Security Through DNS

A holistic approach to designing network architecture and cybersecurity uses DNS for cyber defense to detect attacks at an early stage and fend them off before major damage takes place.

Tools

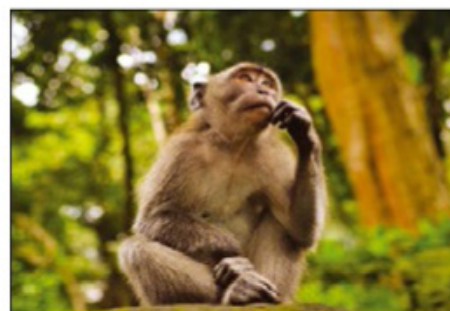
Save time and simplify your workday with these useful tools for real-world systems administration.

22 Apache Mahout

This distributed linear algebra framework delivers new tools and methods for performing data analysis, building machine learning data pipelines, and implementing machine learning models in production.

26 Foundries.io

A modular system for companies wanting to develop Internet of Things applications for devices.



News

Find out about the latest plays and toys in the world of information technology.

8 News

- Hybrid IT leads to complexity and lack of visibility, report says
- 8 admin tasks to automate
- Databricks fully open sources Delta Lake
- Job changes mean higher salaries for Cloud professionals
- Serverless architecture lags in adoption

Tools

32 PowerDNS and MariaDB

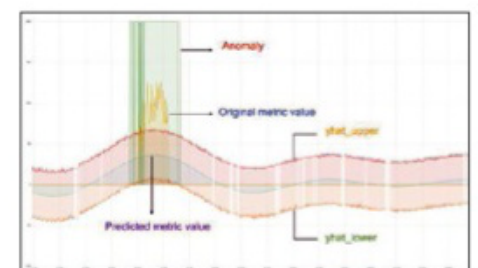
Combining the PowerDNS authoritative server daemon with MariaDB's multiprimary Galera cluster allows a simple yet robust solution for your DNS needs.

36 Dogtag

This certificate manager integrates into the FreeIPA open source toolset to generate SSL/TLS certificates for intranet services and publishes them on the network.

42 Prometheus Anomaly Detector

The Prometheus time series database automatically detects, alerts, and forecasts anomalous behavior with the Fourier and Prophet models of the Prometheus Anomaly Detector.



46 Orchestration with Puppet Bolt

This free software automates administrative tasks to speed up the admin's daily work.

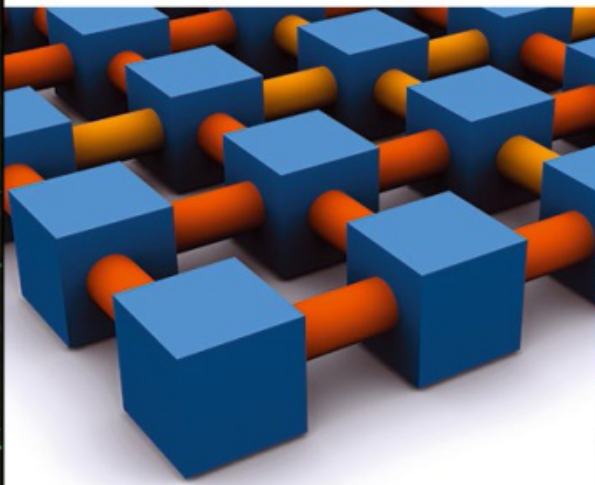
Service

3 Welcome

6 On the DVD

97 Back Issues

98 Call for Papers



- 22 Apache Mahout Distributed Linear Algebra Framework**
Mahout performs high-volume parallel computation on multiple software and hardware systems.



- 68 Azure AD with Conditional Access**
Trust is good; controls are better. Conditional Access sharpens the blurred boundaries that soften models of trust.



- 10-year support life cycle
- PHP 8.0
- Go toolset 1.17.7
- Linux kernel 5.14.0-70
- OpenSSL 3.0.1
- OpenSSH 8.7p1
- SELinux performance improvements
- Automatic configuration of security compliance settings
- NetworkManager key files for new profiles
- GCC 11.2.1
- Go 1.17.7

See p 6 for details

Containers and Virtualization

Virtual environments are becoming faster, more secure, and easier to set up and use. Check out these tools.

- 50 Azure Automation**
A cloud-based service for handling automation tasks, managing updates for operating systems, and configuring Azure and non-Azure environments, with a focus on VM update management and restarting VMs.
- 56 Jaeger**
The various components of cloud-native applications are always exchanging information, which makes troubleshooting difficult. The Jaeger tracing framework helps hunt down the perpetrators.

Security

Use these powerful security tools to protect your network and keep intruders in the cold.

- 62 Teleport**
Centrally manage logins against various protocols, including SSH, Kubernetes, and databases. Functions such as two-factor authentication are included in the scope of delivery, as is management of your own certificates.
- 68 Microsoft Cloud Zero Trust**
Azure AD with Conditional Access makes use of components such as device management, risk assessment, and user roles to create a new mindset for zero trust.

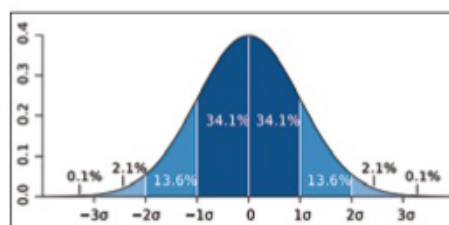
Nuts and Bolts

Timely tutorials on fundamental techniques for systems administrators.

- 72 AD Sites and Services**
Active Directory domains distributed across multiple physical locations with IP subnetting and network configuration allows for replication and universal user logins.
- 78 macOS Shortcuts App**
The automation tool of the iPhone and iPad moves to macOS Monterey 12 to help users make their everyday work more convenient.



- 84 Performance Health Check**
Many HPC systems check the state of a node before running an application, but not very many check that the performance of the node is acceptable before running the job.



- 90 Weka**
This open source tool applies a wide variety of analysis methods to data without the need for advanced programming skills and without having to change environments.

Rocky Linux 9 (x86_64)

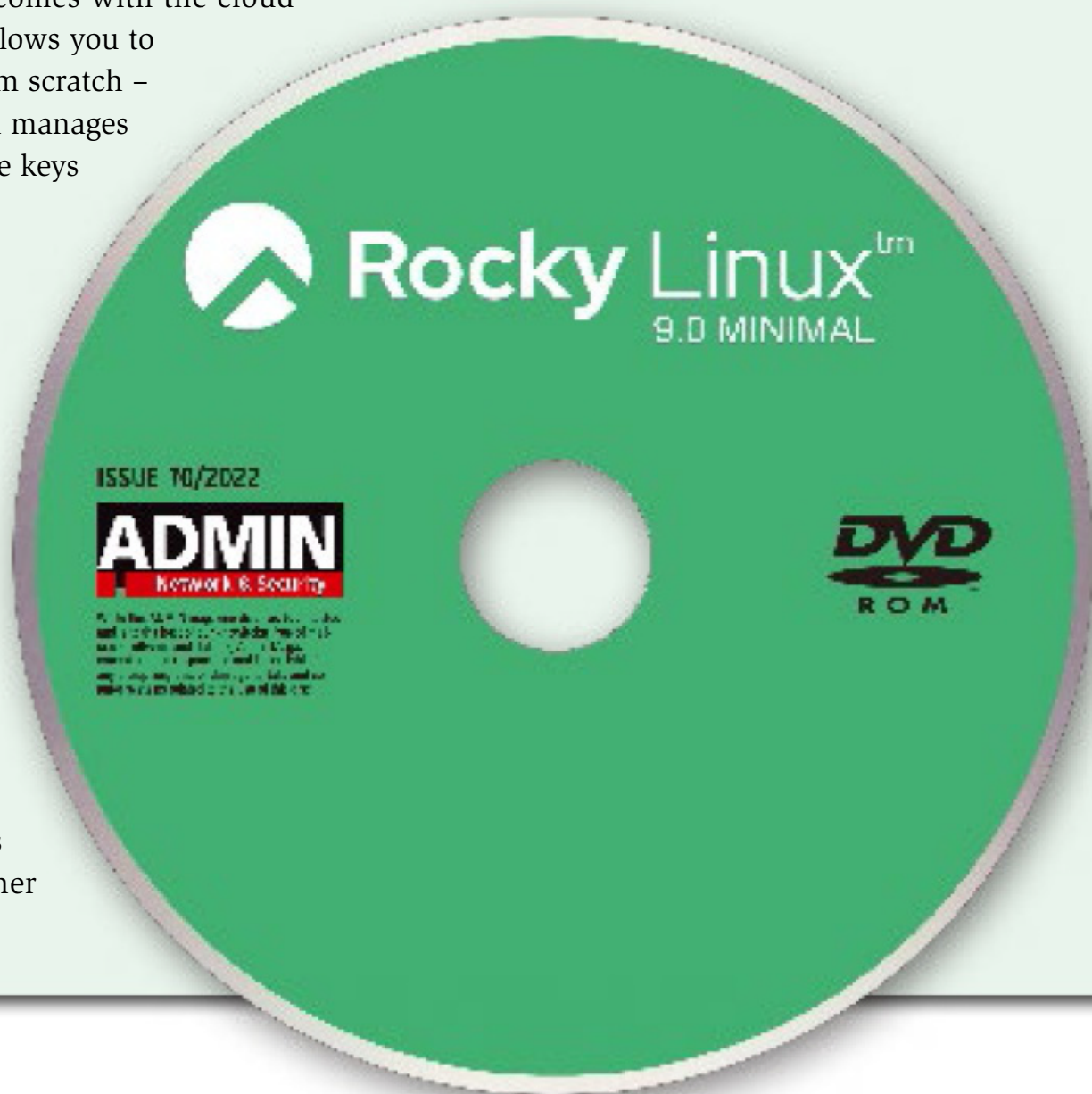
On the DVD

Rocky Linux is an open source, community-developed, production-ready distribution supported by the Rocky Enterprise Software Foundation (RESF). The enterprise operating system is “100% bug-for-bug compatible with Red Hat Enterprise Linux” [1] and has a 10-year support life cycle with regular updates. Rocky Linux comes with the cloud native Peridot build system, which allows you to extend or reproduce Rocky Linux from scratch – thus avoiding end-of-life issues – and manages infrastructure and secure material like keys and secure boot shims.

In version 9, you’ll find:

- Linux kernel 5.14.0-70
- OpenSSL 3.0.1
- OpenSSH 8.7p1
- SELinux performance improvements
- Automatic configuration of security compliance settings
- NetworkManager key files for new profiles
- GCC 11.2.1
- Go 1.17.7

The Rocky Linux team recommends a fresh install of major versions rather than upgrading older versions.



DEFECTIVE DVD?

Defective discs will be replaced, email: cs@admin-magazine.com

While this *ADMIN* magazine disc has been tested and is to the best of our knowledge free of malicious software and defects, *ADMIN* magazine cannot be held responsible and is not liable for any disruption, loss, or damage to data and computer systems related to the use of this disc.

Resources

- [1] Rocky Linux: [\[https://rockylinux.org\]](https://rockylinux.org)
- [2] Build system source code: [\[https://github.com/rocky-linux/peridot-releng\]](https://github.com/rocky-linux/peridot-releng)
- [3] Networking changes: [\[https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/9.0_release_notes/new-features#enhancement_networking\]](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/9.0_release_notes/new-features#enhancement_networking)
- [4] Release notes: [\[https://docs.rockylinux.org/release_notes/9_0\]](https://docs.rockylinux.org/release_notes/9_0)
- [5] Known issues: [\[https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/9.0_release_notes/known-issues\]](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/9.0_release_notes/known-issues)

AKADEMY 2022

Akademy is the annual event for KDE Community members, developers, translators, designers, and friends. Come join us!

BARCELONA
OCTOBER 1-7

akademy.kde.org/2022



News for Admins

Tech News

Hybrid IT Leads to Complexity and Lack of Visibility, Report Says

Nearly half of technology professionals (49%) say the continued adoption of hybrid IT computing is leading to more complexity for IT management, says Sean Michael Kerner, detailing findings from the recent SolarWinds IT Trends Report 2022 (<https://it-trends.solarwinds.com/#/>).

“The complexity comes in multiple forms,” writes Kerner, “including the continued requirement to maintain legacy stacks and the fragmentation between old and new technologies.”

Additionally, 49 percent of survey respondents noted that “visibility into their IT operations for infrastructure and applications has been diminished.”

This lack of visibility and monitoring capabilities leads to organizational gaps, says Kerner, including the ability to effectively conduct anomaly detection and root-cause analysis and to gather metrics from disparate systems.

Learn more at Data Center Knowledge (<https://www.datacenterknowledge.com/cloud/solarwinds-it-trends-report-reveals-hybrid-it-complexity-challenges>).



© Claudia Soraya, on Unsplash

8 Admin Tasks to Automate

Many system administration tasks can be automated to improve your team’s productivity, efficiency, and precision, says William Elcock. For example, manual tasks that admins regularly repeat should be automated to save time and reduce human error.

This article describes eight top tasks that sys admins should consider automating, including:

- Patching
- Password resets
- Disk usage scans

Learn more at ServerWatch (<https://www.serverwatch.com/guides/system-administrator-tasks-to-automate>).



© tomatisch zusammen, 123RF.com

Lead Image © vlastas, 123RF.com



**Get the latest
IT and HPC news
in your inbox**

**Subscribe free to
ADMIN Update
and HPC Update**
bit.ly/HPC-ADMIN-Update

Databricks Fully Open Sources Delta Lake

Databricks has announced (<https://finance.yahoo.com/news/databricks-announces-major-contributions-flag-ship-153000807.html>) that it will contribute the entirety of its Delta Lake storage framework to the Linux Foundation and open source all Delta Lake (<https://delta.io/>) APIs as part of the Delta Lake 2.0 release.

The Delta Lake framework enables building a “Lakehouse architecture” on top of data lakes (<https://databricks.com/discover/data-lakes/introduction>). It provides ACID transactions for concurrency control, scalable metadata handling, and unifies streaming and batch data processing. The new 2.0 release of Delta Lake (<https://github.com/delta-io/delta/releases/tag/v2.0.0rc1>) features improved query performance as well as general improvements for writing large scale performance benchmarks.

Databricks also released MLflow 2.0, (<https://docs.databricks.com/dev-tools/api/latest/mlflow.html>) which includes a new Pipelines feature to accelerate and simplify ML model deployments. The company additionally introduced Spark Connect, which allows Apache Spark to run on any device, and Project Lightspeed, a next-generation Spark streaming engine.



Job Changes Mean Higher Salaries for Cloud Professionals

O'Reilly's 2022 Cloud Salary Survey indicates that changing jobs can result in a significant salary increase — of 20 percent or more — for cloud professionals, reports FOSSlife.

Other results include:

- Survey respondents earn an average salary of \$182,000.
- 20% of respondents reported changing employers within the past year.
- 63% of respondents work remotely full time.
- 94% work remotely at least one day a week.

This year's survey was limited to U.S. participants and compared salary results by state, education level, age, job title, and certifications earned, among other things.

See more results at FOSSlife (<https://www.fosslife.org/job-changes-drive-higher-salaries-cloud-professionals>).



Serverless Architecture Lags in Adoption

Containers and microservices are used three times as often as Functions-as-a-Service and serverless architecture, according to recently published research (<https://www.digitalocean.com/currents/june-2022>) from cloud infrastructure provider DigitalOcean.

“The trifecta of containers, container orchestration systems like Kubernetes, and microservices are commonly used in the workplace, but serverless and Functions-as-a-Service lag behind in adoption by organizations,” reports Lawrence Hecht.

See more at The New Stack (<https://thenewstack.io/serverless-usage-not-popular-in-workplaces-digitalocean-survey-reports/>).





Extended detection and response in networks, endpoint devices, and the cloud

Searching for a Cure

Extended detection and response (XDR) integrates security functions across endpoint devices and networks. But is XDR the only integrated approach to cybersecurity challenges? We investigate the new technology. By Martin Kuppinger

Information technology (IT) is indispensable for core processes in companies that face a tremendous threat to their IT systems. Cybersecurity has moved beyond the IT department to become a central management task. Laws, regulations, and the associated rules of critical infrastructures (CRITIS) make it clear how great is this threat and the need for suitable countermeasures. Manufacturers and service providers have long since responded with an almost countless range of products and services, from traditional software products such as antivirus to artificial intelligence (AI)-based systems for identifying security incidents and the complete operation of security operations centers as a service. One of the biggest challenges is not the lack of suitable technology, but how to use it correctly and the personnel and knowledge required to do so. Even where technology is good and powerful, it still has to be used properly, and the skills gap (i.e., the lack of personnel and knowledge) has long been a central issue, especially in the complex field of IT security.

In this environment, can improved and more powerful integrated solutions such as extended detection and response (XDR) be understood, and what exactly do you need to understand these solutions?

Devices and Networks

XDR as a term emerged in 2018 and is attributed to software vendor Palo Alto Networks. As the term implies, it is about extending existing systems and detecting, identifying, and responding. The integrated approach is not inherent in this term but is an important implicit component. XDR systems are typically offered as software as a service (SaaS), although this is not a requirement in terms of strategy. The extension part in XDR specifically refers to endpoint detection and response (EDR), as well as network detection and response (NDR). XDR now creates approaches that focus on both endpoints and networks, where endpoints are by no means just client systems (e.g., notebooks, PCs,

tablets), but go beyond that to workloads in the cloud. In other words, this definition is genuinely broad, far beyond the scope of EDR and NDR. XDR collects data from various systems and then correlates and provisions the data in a structured manner for downstream analysis. One key part of XDR's functionality is automatic detection of threats, including complex threats that only become visible through an analysis of data across multiple devices and networks. The detected threats are analyzed, sorted, and prioritized so they can then be dealt with in a targeted manner. On the basis of this analysis, it is then possible to react to possible attacks. On the one hand, XDR's value promise stems from its integrated approach, which is designed to detect even complex threats better by correlating data from a variety of different systems. At the same time, vendors tout the benefits of SaaS-based integrated products that are implemented quickly, instead of a multitude of standalone systems that would first need to be linked together. The basic idea behind this process makes sense, especially if you look at the situation in many organizations today, with a large number of IT security products in use as isolated solutions, generating a great deal of overhead in terms of both licensing and operating costs.

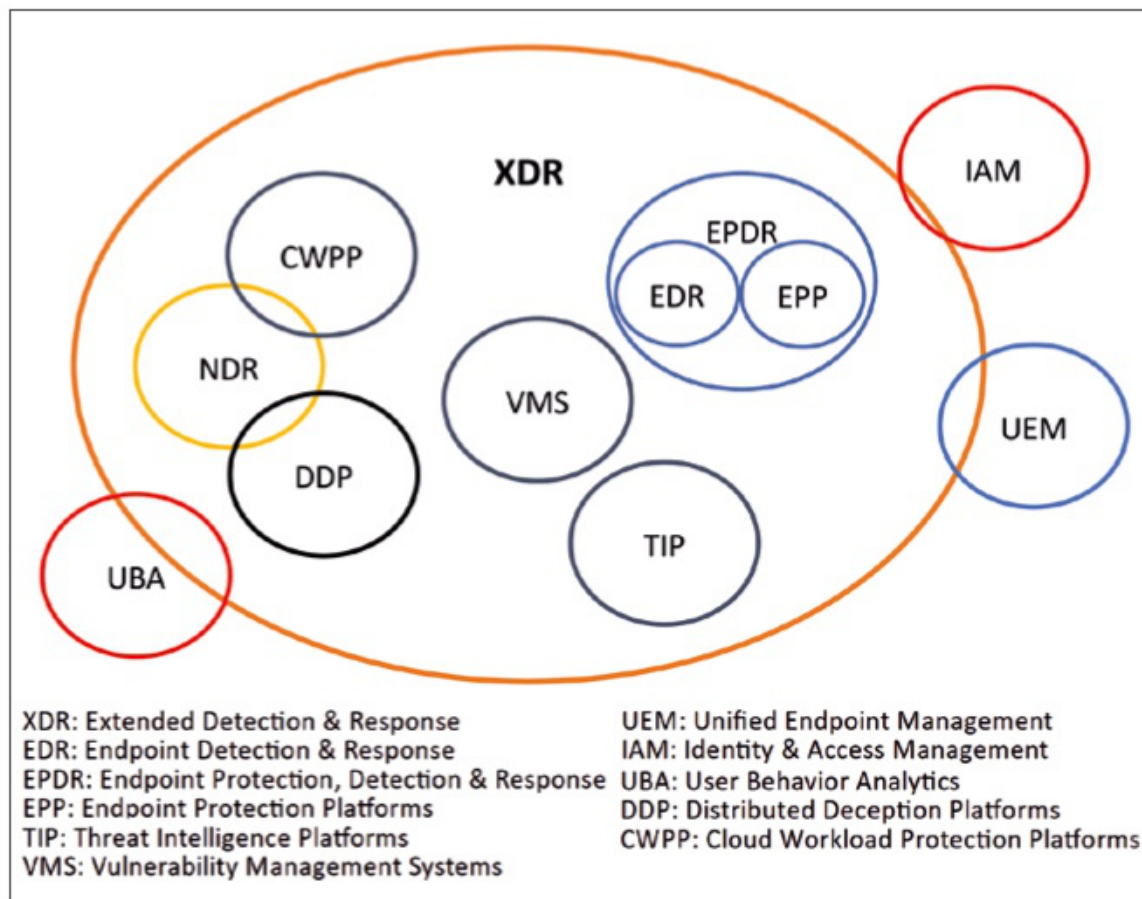


Figure 1: XDR is the combination of a variety of technologies.

Detection and Response

XDR is not a technology for all aspects of IT security. As the term suggests, the focus is on the phases of detection and reaction – or in the more common wordage: response. Currently, the two most popular frameworks for IT security – National Institute of Standards and Technology (NIST) Cybersecurity Framework (CSF) and MITRE D3FEND – address central areas, and the protection side, which has long been in the foreground, is supplemented by XDR. One way to view these technologies is that XDR involves extending protective measures (e.g., firewalls, antimalware, and other solutions) to include continuous analysis of data to detect potential threats and provide a targeted response, and NIST CSF describes the established cycle from risk identification, through protection, to detection, response, and recovery. Various established frameworks and standards such as ISO 27001 are referenced (e.g., to identify and describe the risk areas and protective measures). MITRE D3FEND is clearly more technical in nature and primarily focuses on specific technical measures, with the first phase being hardening the systems, followed by

detection and three detailed response phases: isolation, deception or diversion, and threat elimination.

XDR focuses on the detect and respond phases (i.e., respond, isolate, deceive, and evict). However, for a holistic approach to IT security, the other phases (e.g., as described in NIST CSF and MITRE D3FEND) need to be in place. In addition to this indispensable protection, the ability to recover quickly also plays a central role, especially when faced with ransomware.

Integrated Technology Platform

Like any popular IT technology, XDR is interpreted quite differently by different vendors. A basic understanding exists regarding the combination of network- and end-device-related functions and the correlation and analysis of events across these different domains. In terms of the functions implemented and of the scope of the analytic functions, the solutions have some considerable differences. **Figure 1** provides an overview of core functions and important integrations. As already mentioned, the mandatory functions in XDR are NDR and EDR. EDR products are now typically offered as endpoint protection,

detection, and response (EPDR) or as an endpoint protection platform (EPP). NDR analyzes data from networks; leverages threat intelligence information, including information from external entities; and performs correlations – typically by established static and analytical techniques, as well as machine-learning-based approaches. The goal is always to identify deviating and critical patterns and to derive concrete indications of possible threats and tangible suggestions for possible countermeasures.

Part of the value proposition of XDR is that it identifies potential threats proactively and detects unknown threats (i.e., threats that were not previously known or documented) by anomaly analysis, making it more active than EDR and network traffic analysis (NTA). That said, many of today's EPDR and NDR systems use comparable approaches, just without the integration approach that XDR uses. EPDR works similarly, but with a focus on device usage. EPDR traditionally comes from the client area but, in the meantime, especially in the XDR environment, has developed significantly beyond clients.

Other technologies found in XDR systems include cloud workload protection platforms (CWPPs) for analyzing and protecting functions delivered through cloud services, distributed deception platforms (DDPs) for automating the process of creating sitting duck systems to distract attackers, and vulnerability management systems (VMSs) for detecting vulnerabilities in the IT infrastructure.

XDR also interfaces with user behavior analytics (UBAs) and user and entity behavior analytics (UEBAs) for detecting anomalies in user behavior, with unified endpoint management (UEM) for managing and securing endpoints, along with identity and access management (IAM) for managing users, their authorizations, and, in particular, authentication information that is important in the context of security analysis. Last but not least, of course, is the need to integrate with threat intelligence platforms that provide information on current threats and update it

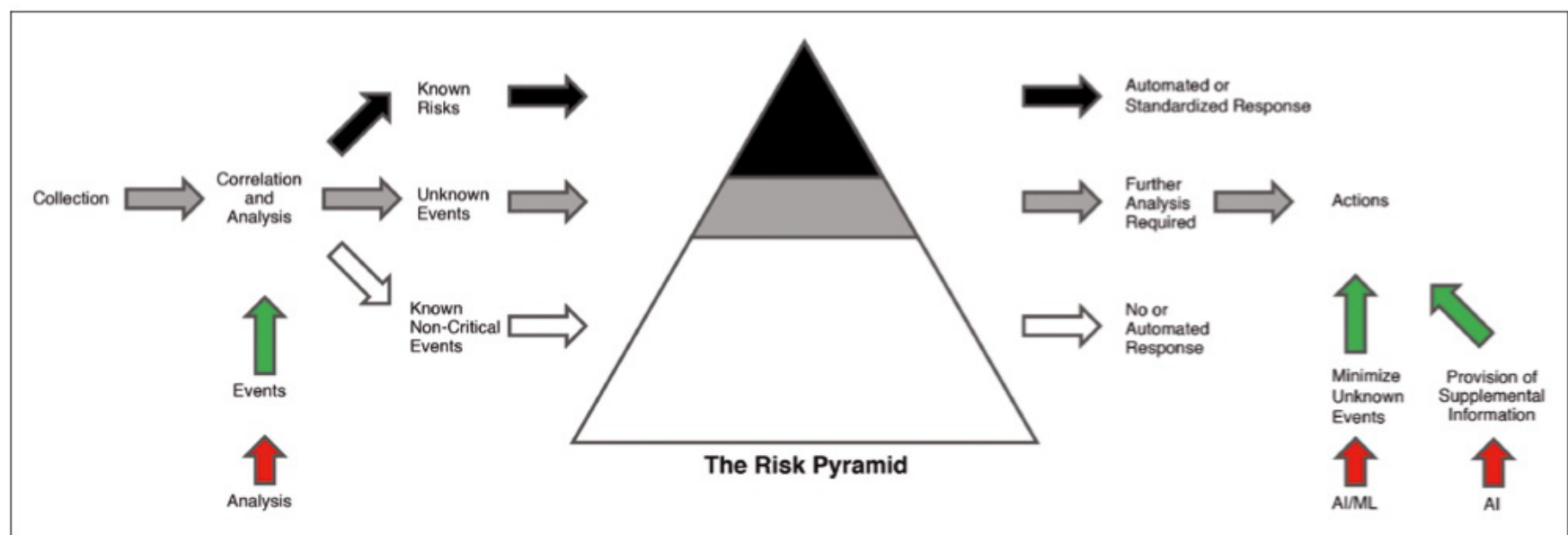


Figure 2: The challenge in IT security is to minimize and target unclear events.

continuously. XDR systems need to adopt this information immediately when analyzing the acquired data to respond quickly to new threats.

Integrated XDR

XDR environments should provide a wide range of capabilities and not just a mere bundling of separate products. Because XDR is typically a SaaS service, functionality should include unified licensing, subscription, and integrated deployment. Centralized dashboards that visualize the threat landscape are also a mandatory requirement for XDR products. Events need to be consolidated across the various networks and systems.

A high level of performance in correlating and analyzing information to provide usable information is a central feature. One of the key challenges in IT security is that, out of the huge number of signals collected, the truly critical threats need to be analyzed to find a response.

The analysis results of XDR (Figure 2) can be divided into three groups. The events shown in black are clear and known threats. Ideally, the response can be automatic (e.g., by redirecting the event to decoy systems by DDP technologies to render the attacks ineffective). The events shown in white are clearly identified as non-critical. In individual cases, an automated response might still be needed, for example, by IT operations management (ITOM). The middle, gray area is the most problematic because it contains

the unknown events that require further analysis by humans. A good XDR solution must reduce this section to the extent possible, while providing good advice as to how to handle the event. One of the essential functions in analytics is the ability to handle encrypted data and either decrypt the data or draw conclusions by reference to the metadata. Additionally, a wide range of analytical capabilities must be available to identify specialized forms of attack, such as an accumulation of unusual DNS requests or an unexpectedly high or low volume of port scanning operations. The broader the analytical capabilities, the more likely complex attacks will be detected.

On the other hand, an XDR system's performance is also determined by the width and depth of the sensors (i.e., the components that collect data on the network or on end devices).

An important component of XDR now

increasingly common in the NDR area is integration with operational technology (OT) environments. Cloud workload protection platforms as an XDR building block provide the interfaces to common cloud environments. XDR is a complex, multilayered technology precisely because it integrates and extends a variety of existing IT security technologies (Figure 3). However, as I mentioned earlier, the functional differences between solutions are significant, which is what makes a thorough investigation indispensable.

XDR, SIEM, and SOAR

One question that continually crops up in the context of XDR is how it relates to and interacts with security information and event management (SIEM) and security operations, automation, and response (SOAR)

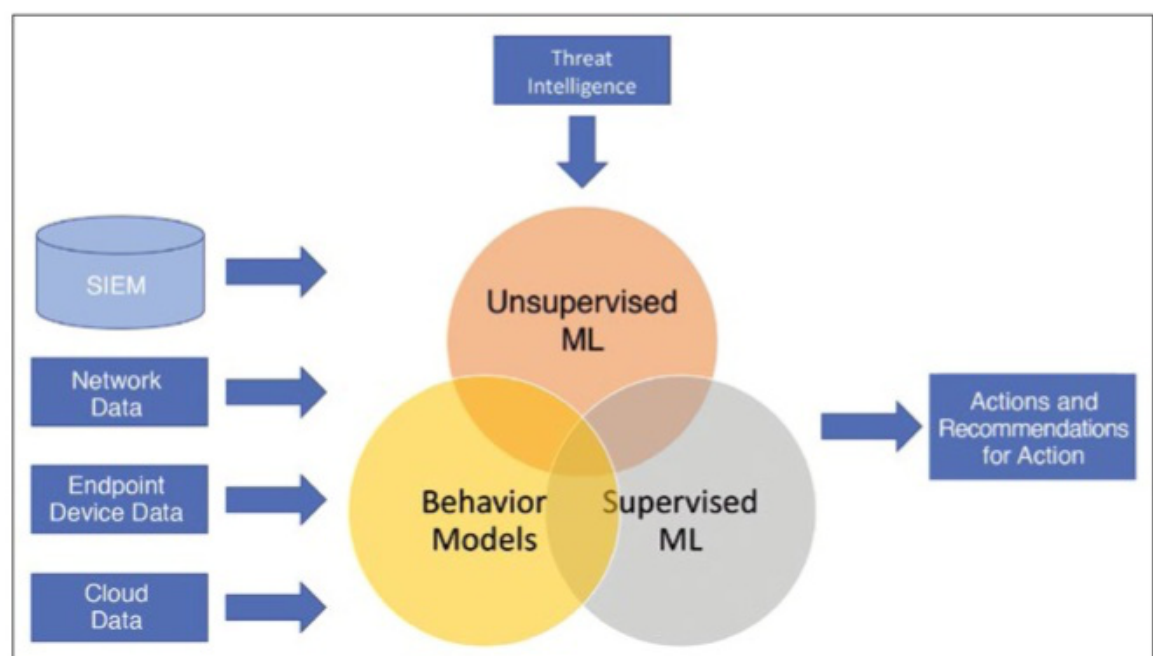


Figure 3: XDR uses a variety of information sources and analytics technologies.

systems. The boundaries between these technologies are fluid. First and foremost, SIEM platforms collect security-related information and events from different systems and other sources. The value proposition of SIEM also includes continuously analyzing this information to act on it. In this respect, SIEM and XDR are closely related, with SIEM typically seen as a source of information for data that XDR systems process, although it can also serve as a target system for information from XDR. Ultimately, however, the precise nature of the interaction also depends on whether a SIEM solution is primarily used as a plain vanilla database or whether analytical functions are also deployed to some extent. SOAR, on the other hand, mainly focuses on operations and the response to threats, with SOAR products also collecting information from a variety of sources. External sources with up-to-date information on threats (threat intelligence) are particularly important. SOAR systems are typically used in combination with XDR and SIEM, with an increasing convergence of SIEM and SOAR emerging. Ultimately, before deciding on the specific solution portfolio, you will need to ascertain which functions are needed, what the priorities are, and which systems are already in place. This requires a portfolio review of the existing IT security products to ensure that you are not just adding one more system, but sensibly integrating a deliberately limited number of solutions. XDR, with its integrative approach, can play a central role in this process. What is also important in this analysis and decision-making process is how the technology will be operated.

Security as a Service

This question brings to the fore the interplay between XDR, managed detection and response (MDR), and security operations centers as a service (SOCaaS). MDR describes an approach in which XDR environments and other systems are monitored by

an external vendor who also responds to any security incidents. MDR is therefore not primarily a matter of technology, but of operations. The same applies to SOCaaS, which involves offerings in which service providers assume responsibility for setting up and operating security operations centers for multiple clients, typically in a defined interaction with internal IT security team staff so that customer-specific requirements and applications can be addressed. Whereas MDR focuses on technical threats, SOCaaS encompasses a broader range of services, including SIEM and SOAR system operations and security technologies such as next generation firewalls (NGFWs). However, a SOCaaS approach also specifically includes incident response management (IRM; i.e., incident preparation and structured security incident response), whereas MDR is primarily focused on threat analysis and response at a technical level. Again, the boundaries are fluid, as is so often the case. Managed security service providers (MSSPs) also have offerings that provide additional services, such as vulnerability assessment, application and code security analysis, penetration testing, IAM operation, and other services.

Self-Operated XDR

For many organizations, the question is whether they are even capable of effectively and efficiently running an XDR environment themselves. In the vast majority of cases, the answer is going to be “No,” because XDR requires a high level of skills and up-to-date knowledge of security threats – even if the application manages to provide concrete, usable threat intelligence. Even then, employees need to understand the intelligence and respond appropriately.

In these cases, cooperation with service providers proves useful, because they can draw on expertise. Whether this takes the form of an MSSP approach, a SOCaaS offering, or simply MDR for the more technical side of

XDR system operations depends on the skill set available within the organization. Moreover, the offered services are never exclusively about the XDR solution.

The decision must always be made within a higher level framework (i.e., with a view to the existing and future overall architecture of the IT security solutions with SIEM and SOAR), which is prerequisite to an organization arriving at both manageable and affordable solutions that focus on helping to identify and address critical risks to the extent possible. Moreover, a holistic concept needs to include the phases that go beyond detection and response (i.e., identification, protection, hardening, and recovery) in case an attack causes damage.


Conclusions

XDR is an interesting and logical development in the technology space because it integrates different technologies in a meaningful way to provide a holistic view of security threats. Before you look into XDR, however, you first need to define an overall concept that includes both the technical architecture and modular solutions to be deployed and, in particular, the operating concepts. Without such an overall picture, XDR is just another isolated solution that fails to deliver the promised value in terms of IT security improvements.

Additionally, XDR’s integrative approach always involves focusing on your choice of solution provider (i.e., the risk of dependence on the vendor). Interfaces to other products and strategies that enable a change of provider therefore also need to be taken into account from the outset. In any case, organizations need to review the status of their IT security organization and infrastructure regularly, including analyzing if and where technologies such as XDR, MDR, or SOCaaS can help them reduce threats. ■

Author

Martin Kuppinger is the founder of and Principal Analyst at KuppingerCole Analysts AG.



Hone your skills with special editions!

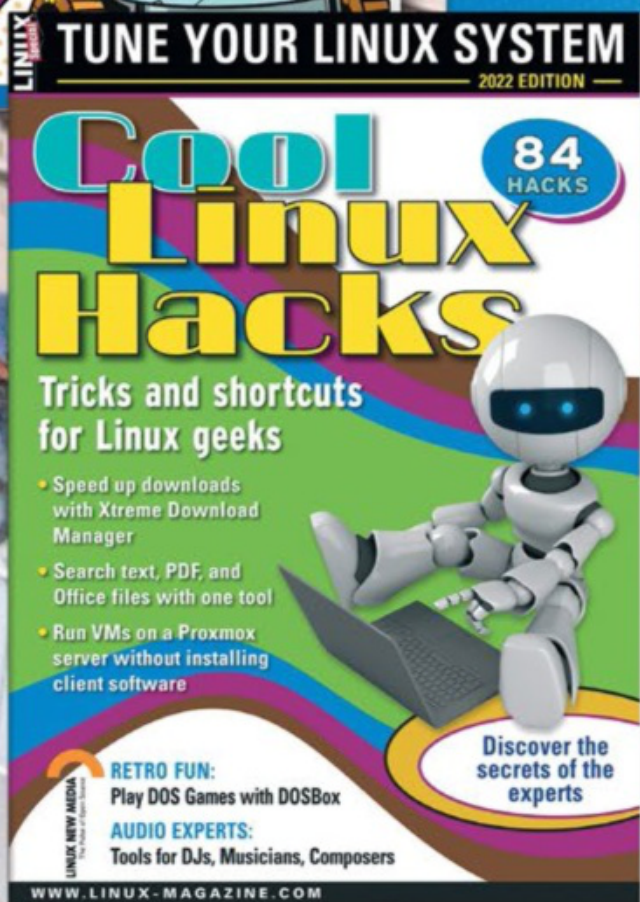
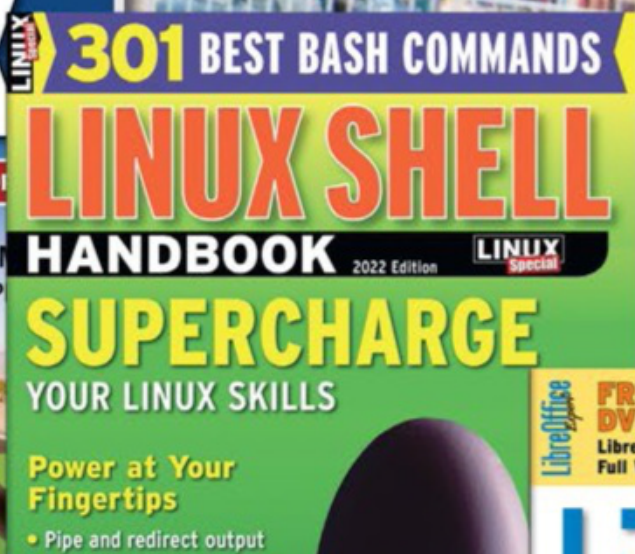
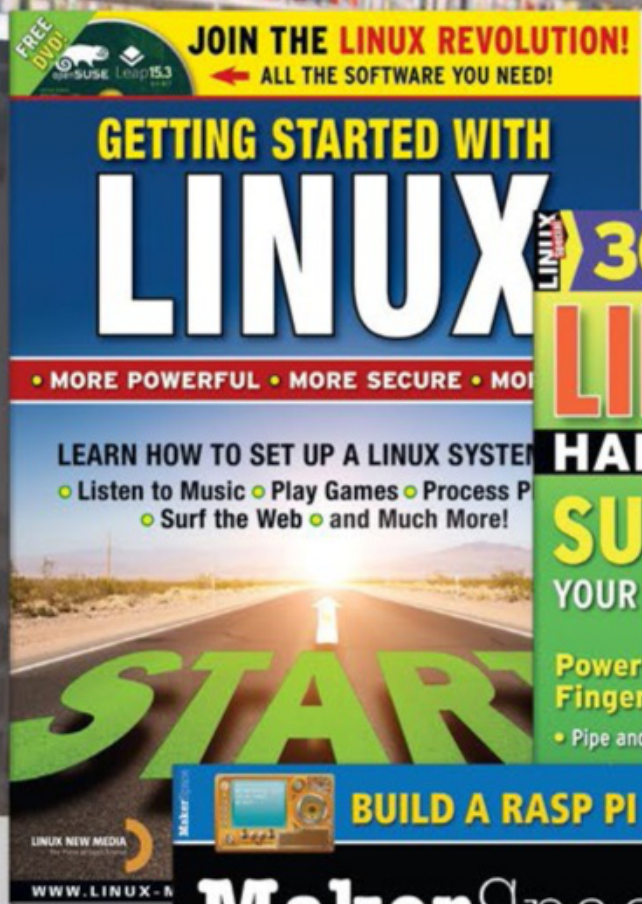
Get to know Shell, LibreOffice, Linux, and more from our Special Edition library.

The *Linux Magazine* team has created a series of single volumes that give you a deep-dive into the topics you want.

Available in print or digital format

Check out the full library!

shop.linuxnewmedia.com



Preparing for cyberattacks

The Enemy in My Web

The possibility of a ransomware attack means it is essential to prepare for cyberattacks by putting defense mechanisms and contingency plans in place. By Matthias Wübbeling

The number of cyberattacks with ransomware has been rising steadily for several years. WannaCry ransomware attacks caused quite a stir in 2017. Hundreds of thousands of Windows systems were infected through a vulnerability in Microsoft's SMB

protocol, and the data on these systems were encrypted. The malware used a US National Security Agency (NSA) exploit named EternalBlue published for propagation by a hacker group. Although Microsoft released a patch to close this gap before the

WannaCry outbreak, many systems had not yet been updated and were therefore still vulnerable (**Figure 1**). More or less by accident, British security researcher Marcus Hutchins found a way to disable WannaCry. The malware checks for the existence of a special domain before encrypting files. If the domain is not accessible, WannaCry starts encrypting. After the registration of this domain in the worldwide DNS system, further propagation was temporarily stopped after just four days. By then, Bitcoin payments equivalent to several hundred thousand dollars had already been transferred to the attackers' wallet.

The encryption of more than 30 servers on the computer network of University Hospital Düsseldorf in the fall of 2020 by a modified WannaCry variant attracted a great deal of attention in the German media. Because IT was unavailable at the hospital, one patient likely died because she could not be admitted and had to be transported to a hospital further away.

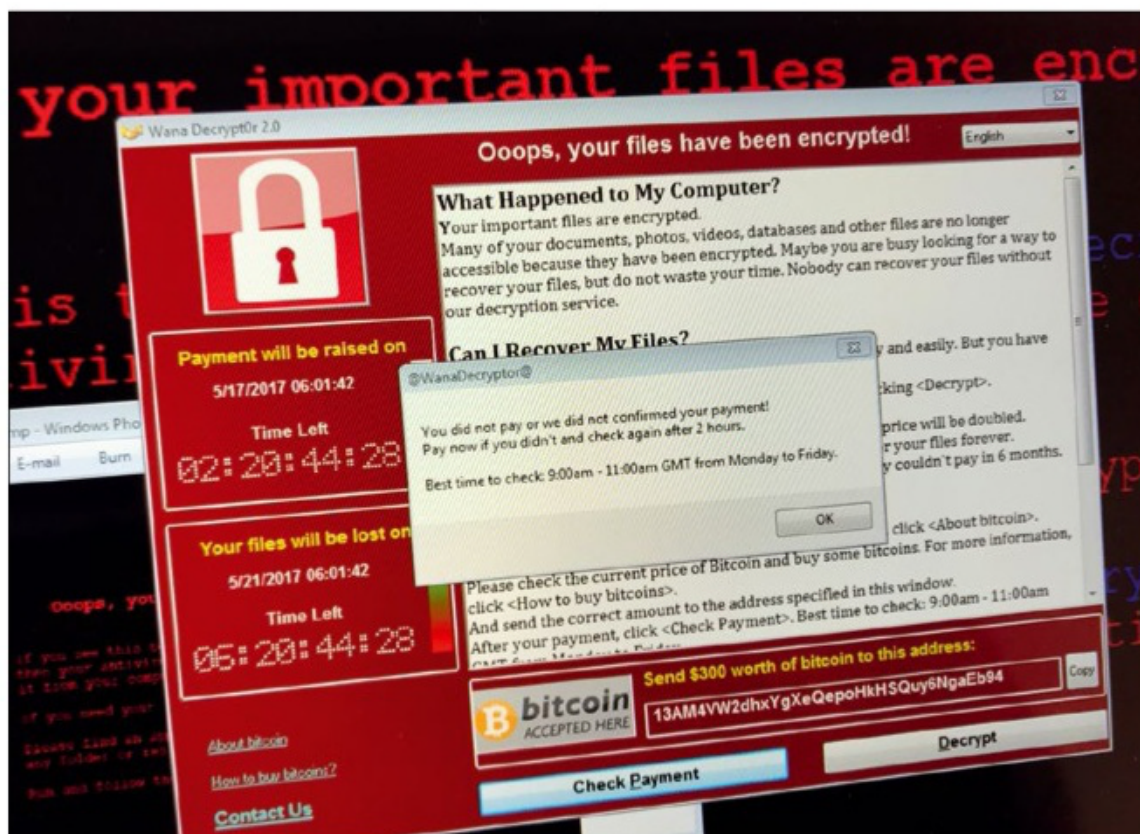


Figure 1: Game over. If you see a message like this on the screen, ransomware – in this case WannaCry – has struck.

Access Vectors

In most cases, the goal of ransomware is to extort a ransom. Emotet [1] and EternalBlue [2] are just two of many ways criminals take control of computers and encrypt existing files. The Emotet botnet was dismantled by Europol investigators in early 2021 and the infrastructure was shut down. However, this huge success on the part of the European authorities only alleviated the threat situation for a short while. The first modified Emotet variants were already in circulation by November 2021.

In addition to email and security vulnerabilities, leaked employee login credentials offer a regular gateway into corporate networks, as illustrated by the Colonial Pipeline case in the US [3]. Attackers apparently found valid account credentials for an employee on the Darknet, accessed the company's computer systems over a virtual private network (VPN), and installed ransomware. As a result of the encrypted files and the ransomware, Colonial shut down the entire pipeline system for more than five days. The resulting shortfall in fuel supply led to a sharp rise in fuel prices in some parts of the US.

Colonial paid the equivalent of nearly \$5 million in ransom to the extortionists. However, the extortionists did not simply rely on encrypting Colonial's files. Instead, they copied almost 100GB of files from the Colonial systems beforehand. The blackmailers additionally threatened to publish this information if Colonial did not pay the ransom. This practice of copying files before encrypting them opens another attack vector. Affected companies will still pay even if the data can be easily restored from a backup to prevent the publication of internal data and possible company secrets.

Perpetrator and Victim

The perpetrators of large ransomware incidents are mostly well-organized

groups. After postmortem investigations and publications, the hacks are repeatedly attributed to hacker groups in Eastern Europe or Russia. This attribution primarily relies on analyzing the malware and the perpetrators' communication with their victims, but small groups of perpetrators also use ransomware, although they do not develop it themselves. Ransomware construction kits can be found on various forums for a small sum.

If you look at the public coverage of ransomware incidents, you can't help feeling that it is mainly large institutions and public bodies that are targeted. However, this image is deceptive. Small and medium-sized enterprises (SMEs) are just as much victims of ransomware as are private individuals. SMEs in particular sometimes suffer massive damage after cyberattacks. No official figures indicate in how many cases the ransom was paid; however, the continued high number of attacks is an indication that the method is successful.

Although the perpetrators are often not choosy in their victims, they do take a targeted approach after a successful raid by analyzing the network infrastructure so they can hijack as many systems as possible. Moreover, backup systems are identified to make the recovery process more complicated. Only after the attackers have grabbed the sensitive data does the encryption process begin, with the ransom demand following on its heels.

Although private computers might not seem particularly worthwhile targets at first glance, the consequences can be explosive – even if the owner does not pay a ransom – because the attackers also collect passwords from the usual password safes of the browser or email program. Although these passwords are often secured by a master password, the master can be sniffed. In many cases, access credentials for other computer systems are also preserved in these records, which is how criminals repeatedly find login data for remote maintenance access. The perpetrators then use the data to

Cyber Insurance

If you want to protect yourself against the consequences of cyberattacks, you can resort to the classic means of insurance and take out cyber insurance. Depending on the industries in which you operate, cyber insurance can be a beneficial addition to your risk and contingency plan. Insurance companies can also help you take initial steps after an attack has occurred and refer proven incident analysis partners. Whether or to what extent the consequences of the attack are covered and whether the insurance company will pay a ransom depends on your individual contract. However, some major insurance companies announced last year that they would no longer pay ransom to criminals.

access corporate networks, which in turn means they have found another victim for their malware.

Protections

Different measures are required to protect yourself successfully against ransomware and the possible consequences of an infection (see the "Cyber Insurance" box). Technical vulnerabilities, such as WannaCry and EternalBlue, can be removed by regularly updating all systems in the company. With regard to automated updates, you first need to ask yourself for each system whether a potential failure because of an error during the update process outweighs an existing security vulnerability. You will want to enable automatic updates where possible, even at the risk of failure because of a failed update. The longer it takes to review and release an update, the more time attackers have to access enterprise systems.

You will want to use centralized anti-virus and application layer gateways to scan email attachments before they are delivered to employee accounts. At least the malware variants created by construction kits can often be detected in this way, although they often do not work against the individual variants of the larger groups. To prevent retroactive loading of malware after an infection, you can redirect your employees by web proxies,

prohibit binary downloads or enable them separately, and initially block all other requests to pass by the proxy in the packet filter. Security product providers offer lists of IP addresses and domains that you need to filter for this purpose. These measures often cannot be implemented without restricting employees, and they sometimes interfere with daily work. Therefore, many companies do without them. However, it makes sense to work with other staff to see which measures you can implement. You can also take into account the times when no one usually works. When your office is closed, you can implement and monitor far stricter rules, which are then relaxed again during normal business hours.

Even though a company's users are always portrayed as the highest risk vulnerability, they are really your last line of defense and can prevent an infection, unlike the industrial security products such as antivirus programs, application layer gateways, and special firewalls.

Attackers use spearphishing techniques and, as in the case of Emotet, existing communications to trick users into running the malware. Seeing through these perfidious techniques is difficult even for well-educated and trained employees. Educating your users needs to be an integral part of your overall IT security strategy. Targeted training (including an active error culture and an option to report conspicuous activities) sensitizes your users so that they do not involuntarily help the attackers.

Additionally, you can provide technical support to your employees and establish email signing in your organization to make it at least a little more difficult to create credible email. If all the email in your organization is signed, those messages that aren't will stand out. The fewer exceptions, the more reliably your employees can detect fake email. However, if a user opens a malicious attachment, you need to protect the system with active group policies that prohibit the execution of macros in these files. If users need macros for their daily work, then at least use

signed macros and prevent unsigned macros from running.

Restricting Access

Valid login credentials in the hands of criminals, in addition to technical vulnerabilities, are a major problem for the security of your computers and services. On the Darknet, hackers can obtain extensive collections of identity data and login information. Because users tend to use the same passwords for different services, you might also be at risk if other services fall victim to hacking. In fact, more than two-thirds of users continue to use previously leaked login data for more than a year. The US National Institute of Standards and Technology (NIST) and the German Federal Office for Information Security (BSI) IT basic protection compendium (IT-Grundschutz-Kompendium) point out the dangers and recommend regular checking of user accounts and passwords.

Different service providers offer identity leak checkers. The free American service *Have I Been Pwned* (HIBP) [4] is probably the best known provider of leak information. By entering an email address, you will receive information about whether it was part of a data leak. Even if it is your company email address, the use of HIBP is questionable for data protection reasons in some countries and should be discussed with the corporate legal department. Moreover, HIBP does not give you direct access to the affected password to this account, so you cannot check it directly against your systems. However, specialized service providers on the market also implement General Data Protection Regulation (GDPR)-compliant checks of login data.

When assigning user rights, you need to consider the possibility of stolen login data and give employees only the access rights they absolutely need for their normal workday, especially when accessing servers and shared files. For example, if you grant users read-only rights for existing files on a server, the users need to upload a

new file for each change, but the user cannot delete or encrypt the files.

The principle of least privileges means, above all, that you need to establish processes that regularly check the existing privileges, especially when employees change departments or collaborate across departments on projects. The phenomenon is common, for example, among interns who pass through different departments in the course of their internship. Once granted, privileges are often not revoked, but new ones are regularly added. At the end of an internship, the intern then has access to a user account with many security-related access options.

Preventing the Spread

If attackers do gain access to computers in your corporate network, despite all protective measures, this does not necessarily mean that they will ultimately be successful with their attack. Try to mitigate the damage in these cases. To prevent the spread, it makes sense to isolate different departments and different teams in the same department from each other in terms of network technology and locate them on their own subnets. Between these subnets, you need to have a firewall that regulates interdisciplinary network traffic, limited only to what is necessary.

The faster you react, the greater your chance of averting a major loss. Comprehensive monitoring of your resources identify and isolate affected systems quickly. You might want to isolate an entire team or department together. In this way, you can uphold the ability to work and protect the other organizational units in the meantime. Of course, you also need to go through this process regularly. Often, only a few firewall rules are required. Depending on your infrastructure setup, you can also automatically isolate affected computers in a separate virtual local area network (VLAN). An attacker then still has access to a system but cannot infect any other computers from there. If you log internal network connections on the

routers between your departments, you can even determine afterward whether propagation – also known as lateral movement – has taken place. Although attackers might have penetrated your corporate network through a vulnerability, it doesn't mean they will find the same vulnerability on other systems. Attackers therefore use different tools, including the remote desktop protocol (RDP) supplied by Microsoft. Especially in these times of home office and VPN connections from home to the corporate network, remote desktop connections are enjoying great popularity. In most cases, access is quickly granted by Active Directory. Again, you need to monitor connections established with the directory service at central locations and automate your response to undesirable connection attempts to the extent possible.

Protecting Backups

Creating backups is one of the administrator's standard tasks. However, you should not only handle and monitor how backups are created, but also how existing backups are protected and accessed. At best, you have no access to the backup system. Instead, the backup system needs access to the individual services it is supposed to back up. If the users of your systems cannot access the backup themselves, it cannot be encrypted by ransomware launched from a normal user account. To support easy recovery of files for normal use, you will want to establish different backup systems: one that your users can manage themselves, and one that can only be accessed in extreme emergencies and only by a few administrators. Although this action does not protect you against an attacker demanding a ransom to protect your sniffed trade secrets, you can quickly resume operations after a ransomware incident.

Regulating Processes

If the cat is out of the bag and your systems have been affected by a

ransomware incident, you need to respond adequately. Ideally, you will have drawn up risk and contingency plans in advance and defined responsibilities. The plans include information about the criticality of individual systems and specify the extent to which you need to shut down other systems. A targeted and planned shutdown can protect your company against existential damage, even if collateral damage has to be accepted in the process. The operators of Colonial Pipeline responded in an exemplary manner and specifically removed the system from the network.

You need to inform contacts in the affected departments in good time and make sure that backup systems go online in a way that reflects the established criticality. If you are legally required to report cyber incidents, you should have appropriate forms pre-filled and make a report in a timely manner. This procedure will help you avoid penalties because you left steps out. Keep the affected systems as-is for later forensic analysis. You can handle this step yourself if your company is large enough and you have appropriate skills in your IT department; otherwise, commission an external service provider to perform the analysis. The main goal is to identify the vulnerability – one hopes you have been able to restore your data from backup. Gradually rebuild your infrastructure once you have eliminated the vulnerability. While you're at it, don't forget to set up new backup systems to cushion the effect of a new attack.

In the best of all worlds, you will also have an internal contingency plan for each department that will inform suppliers, partners, or customers in the respective areas. If you are a supplier yourself, you need to notify dependent companies in the supply chain in a timely way and inform your own suppliers in these times of zero-stock supply chains and just-in-time production.

If you were caught off guard by the attack while you are still in the

process of working out your risk and contingency plans, at least try to recover what can be salvaged, including consideration of a ransom payment. However, you should arrange this in collaboration with the authorities you notified after the incident. Set up a crisis team with all the people you can identify as relevant in a timely manner and discuss the necessary measures.

Conclusions


Ransomware is a big threat to businesses, public institutions, and individuals. In recent years, the consequences of ransomware attacks have grown in scale. In this article, I looked at the various attack vectors and manifestations of ransomware from actual incidents and discussed the risk that exists and how contingency plans can help you restore operations when responding to attacks. ■

Info

- [1] Emotet: [\[https://en.wikipedia.org/wiki/Emotet\]](https://en.wikipedia.org/wiki/Emotet)
 - [2] EternalBlue: [\[https://en.wikipedia.org/wiki/EternalBlue\]](https://en.wikipedia.org/wiki/EternalBlue)
 - [3] Ransomware attack on Colonial pipeline: [\[https://www.bloomberg.com/news/articles/2021-06-04/hackers-breached-colonial-pipeline-using-compromised-password\]](https://www.bloomberg.com/news/articles/2021-06-04/hackers-breached-colonial-pipeline-using-compromised-password)
 - [4] Have I Been Pwned: [\[https://haveibeenpwned.com\]](https://haveibeenpwned.com)
-

The Author

Dr. Matthias Wübbeling is an IT security enthusiast, scientist, author, consultant, and speaker. As a Lecturer at the University of Bonn in Germany and Researcher at Fraunhofer FKIE, he works on projects in network security, IT security awareness, and protection against account takeover and identity theft. He is the CEO of the university spin-off Identeco, which keeps a leaked identity database to protect employee and customer accounts against identity fraud. As a practitioner, he supports the German Informatics Society (GI), administering computer systems and service back ends. He has published more than 100 articles on IT security and administration.



Employing DNS in network security

Revealing Traces

A holistic approach to designing network architecture and cybersecurity uses DNS for cyber defense to detect attacks at an early stage and fend them off before major damage takes place. By Steffen Eid

The corporate network has long ceased to be a single perimeter with branch offices connected to the outside world by the Internet. In the growing network jungle, however, an overall perspective is often difficult to maintain, which is why dividing the network into individual silos to give it structure seems tempting at first glance. This approach would definitely be wrong, because thinking in silos causes problems. Most important is the often missing ability to communicate between isolated solutions because a wide variety of security tools are implemented in the silos – and usually more than one.

Next-generation firewalls, web gateways, email security, endpoint security – the security solutions in the individual sectors are often piled up on top of one another. The unintended consequence of this strategy is that communication between the individual systems is poor, and often even incorrect. For example, if interfaces are not configured correctly, the security tools can trigger false or duplicate

alerts among themselves, overwhelming what are already overburdened security teams. However, the tool for achieving a unified, comprehensive view of your network already exists – the Domain Name System (DNS). After all, as the hub of communications on the Internet, DNS can be the heart of integrated network management and security.

More Is Not Always Better

In IT departments, when workflows are not fully covered by just one security tool, communication interfaces need to be kept as up-to-date as possible at all times, and employees need to be constantly trained in the use of the many tools. These resources could be put to better use elsewhere. This problem is even more pronounced in large enterprises, which can be geographically widespread and might be working on restructuring such as mobile use, a multicloud rollout, or software-as-a-service (SaaS) and software-defined (SD)-WAN implementations.

According to a Ponemon study sponsored by IBM [1], it still takes more than 280 days on average for a security breach to be detected, but containing a breach in under 200 days would save \$1 million in costs.

Vulnerable Without Safeguards

Without DNS, any activity on the web would be messy because DNS converts the input from URLs into the significantly more difficult to remember IP addresses, helping users access the desired websites. As convenient as DNS is for users, it can also be dangerous if it is not secured properly, because attackers use DNS to communicate with their targets and for data exfiltration. Whether the attack is meant to steal confidential company data (exfiltration), infiltrate with malware in small data packets (infiltration), or create separate communication tunnels to make transferring data even more convenient, hackers use DNS as an access vector into networks.

Photo by Abbas Tehrani on Unsplash

To understand how DNS can help you comprehensively secure your own networks, you need to look back to the early 2000s when DNS security tended to be a minor concern. At the time, the Berkeley Internet name domain (BIND) servers – still an important standard in DNS today – had only two security features: They did not accept responses from IP addresses they had not queried (also known as Mars responses), and they inserted a random 16-bit number into outgoing requests and checked that the number came back in the responses. Only later did analysts discover that this test number was not really random. Little wonder, then, that DNS servers have long been a worthwhile target for attacks, such as the Li0n worm, which exploited a vulnerability in BIND. Moreover, DNS servers of all kinds are often used as amplifiers in distributed denial of service (DDoS) attacks and are still the target of such attacks to this day.

That said, over time, the security of DNS servers and DNS itself has improved. BIND has been optimized to support access control lists for almost everything: queries, recursive queries, zone transfers, and dynamic updates. The DNS community started to operate DNS servers in chrooted environments according to the principle of least privilege. Additionally, transaction signatures (TSIGs) and DNS security enhancements (DNSSEC) were introduced to further protect DNS. Even if DNS itself is not attacked, though, it remains the communication highway that hackers still use for their attacks.

DNS as the First Line of Defense

Cyberattacks are as varied as the attack vectors available to hackers, but almost all of them have one thing in common: They depend on DNS for almost all communication on the network. For example, more than 90 percent of malware uses DNS to exfiltrate data, redirect

traffic, or communicate with the attacker in some other way. Conversely, DNS contains all the data you need to detect an attack. Defenders who keep an eye on their DNS at all times can take advantage of this fact, quickly detect unexpected atypical communications, and take countermeasures. Without question, this task is mammoth. Artificial intelligence helps keep track and automatically filter out harmful communication requests.

The potential of DNS as a security tool in its own right has only recently been recognized. The advent of response policy zones (RPZs) in 2008, meant that DNS servers could be leveraged to issue “benevolent lies” when they received an information request whose response could be damaging to the querying entity. At least as important was the ability to detect when a DNS server was queried for data known to be corrupted. Since then, companies have appeared that prepare DNS threat data in the form of RPZs and offer their customers this data commercially. Organizations can incorporate a variety of RPZ feeds into their DNS infrastructure and enable their DNS servers to protect users and systems against known malware propagation sites, command-and-control infrastructures, and much more.

RPZs are also helpful in centrally monitoring network health, such as detecting infections and security breaches across the board. A laptop that sends a query to a domain name that is clearly used by a certain type of malware is almost certainly infected with that malware. Armed with this knowledge, important measures can be taken quickly and efficiently without the need for one of the many other security tools to sound the alarm first. The benefits of central DNS as a security layer go even further: Organizations that archive all of their DNS query logs have an important tool at hand in the event of an infection. Even if the attacker is not immediately detected, these logs can be used to trace which other systems

the hacker accessed and how they moved around the network. In this way, the attack can be reconstructed and retraced holistically.

Automated Counter to Hackers

The most advanced companies and vendors feed DNS telemetry data – also known as passive DNS – into data stores and then have the data analyzed by machine learning algorithms. Sophisticated algorithms can detect various types of malicious activity in passive DNS data, including, for example, requests sent by a domain generation algorithm (DGA), which is code that automatically creates a list of domains used by malware clients to communicate with a number of command-and-control (C&C) sites.

These domains serve as a meeting point for malware- and hacker-controlled servers that communicate secretly over a backhaul network. Once one of the DGA domains is detected and blocked by IT security, the malware client and C&C server move to the next domain on the list to bypass the defenses. For example, the defense algorithm can detect patterns in the newly created domain names and directly identify them as threats.

Conclusions

DNS is an indispensable part of any modern security toolkit, playing both an active and a supporting role in securing networks and tracking malicious activity. Moreover, DNS is a central tool already in place connecting all departments, which can facilitate the paradigm shift away from silos and toward a holistic integrative approach. ■

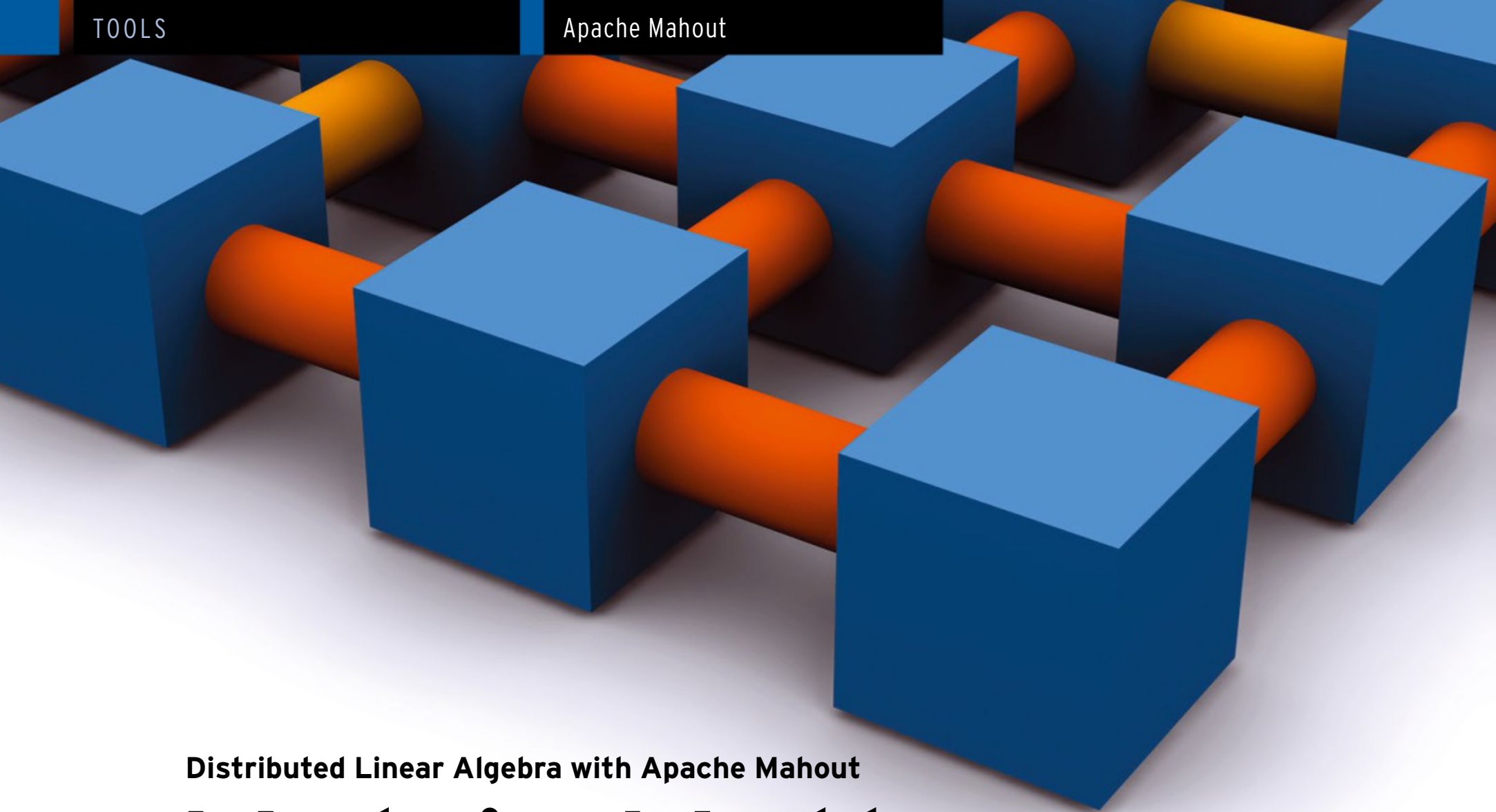
Info

[1] Cost of a Data Breach Report 2021:

[<https://www.ibm.com/security/data-breach>]

Author

Steffen Eid is a Manager for Solution Architects in Central Europe at Infoblox.



Distributed Linear Algebra with Apache Mahout

Matrix Math

The Apache Mahout distributed linear algebra framework delivers new tools and methods for performing data analysis, building machine learning data pipelines, and implementing machine learning models in production.

By Andrew Musselman and Trevor Grant

The ideal scenarios for using Apache Mahout are in teams with the flexibility to adapt as their needs change over time. Mahout can easily swap back-end compute engines (e.g., batch or micro-batch systems such as Apache Spark) or streaming systems (e.g., Apache Flink). Additionally, Mahout is able to perform compute on multiple software and hardware systems, ranging from the Java Virtual Machine (JVM), to whatever multicore CPU is available, to on-board GPU for high-volume parallel computation.

Typical users of Mahout are advanced data engineers and data scientists who have experience writing transformations and models in other languages but who are looking for an approach that does not require they rewrite their work, depending on the system they use from month to month or year to year. Moreover, mathematics-oriented software engineers will enjoy the simplified Samsara domain-specific language (DSL), which provides a stripped-down syntax that feels natural to people accustomed to writing

with math notation, avoiding the usual “syntax bloat” that most machine learning libraries require.

Apache Mahout - What Is It?

Apache Mahout is a library designed to make composing and maintaining distributed linear algebra algorithms easy. First, it creates an abstraction layer on the underlying engine (the open source version of Mahout uses Apache Spark as an engine), and the abstraction layer implements basic linear algebra functions on datasets in the engine (e.g., by defining a distributed matrix, matrix multiplication, multiplication with self transposed, and other functions). Second, it uses Samsara DSL in Scala, which allows users to define algorithms with an R-like syntax that makes it much easier

to write, and later read, complex mathematical formulas. Mahout has a rich history and was one of the (if not *the*) original machine learning libraries for big data. Originally designed to aid in machine learning tasks on data in Apache Hadoop clusters, it underwent a major refactoring around 2014-2015 that resulted in its current form. One challenge this restructuring creates is the abundance of information in circulation that refers to the “old” Hadoop-based Mahout. Machine learning sometimes refers to a set of techniques for numerically solving problems that are too large for standard statistical approaches. Mahout gives you tools to solve those problems with tested statistical approaches. Mahout can be applied to any “big data” platform (e.g., any platform on which

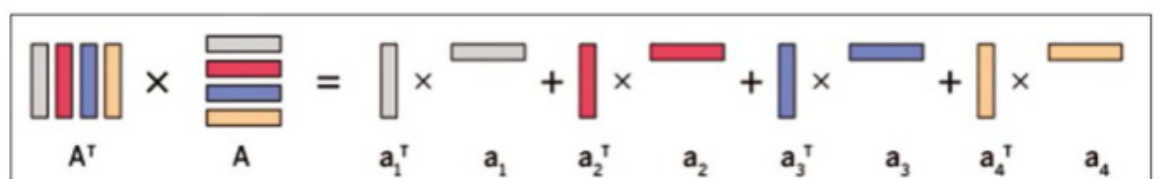


Figure 1: A conceptual illustration of the way Mahout approaches matrix multiplication of matrix A with its own transform.

users are running distributed datasets) and can pay dividends if your company has critical algorithms of concern for refactoring into machine learning approaches.

Distributed Linear Algebra

Linear algebra, otherwise known as matrix math, is a rich and established field of theoretical and applied mathematics that has found applications across multiple spheres of computer science and software engineering, including visual simulations, audio analysis, and predictive analytics. In some cases it can be arithmetic-heavy, with methods built up in iterative or bulk activity, and at scale these computations

can become either overly cumbersome or complicated to program on distributed systems. Fortunately, Apache Mahout abstracts away many of the complexities and pitfalls of distributed linear algebra, leaving a tidy library for working with distributed linear algebra as though it were simply normal linear algebra. For example, in [Figure 1](#), the original matrix A and its transpose A^T are sliced into independent and corresponding rows and columns, which then can be sent across to a compute engine to perform small chunks of arithmetic. The results are then compiled together for a single output. In mathematics, complex procedures are normally the product of many simpler procedures. Mahout takes advantage of

this by introducing an engine abstraction layer, in which someone who is an expert in the underlying engine will implement performant ways to do basic operations like multiplying two matrices, multiplying a matrix times itself transposed, and other operations.

[Figure 2](#) shows the application layers at the top, with multiple engines transparently managing the distribution

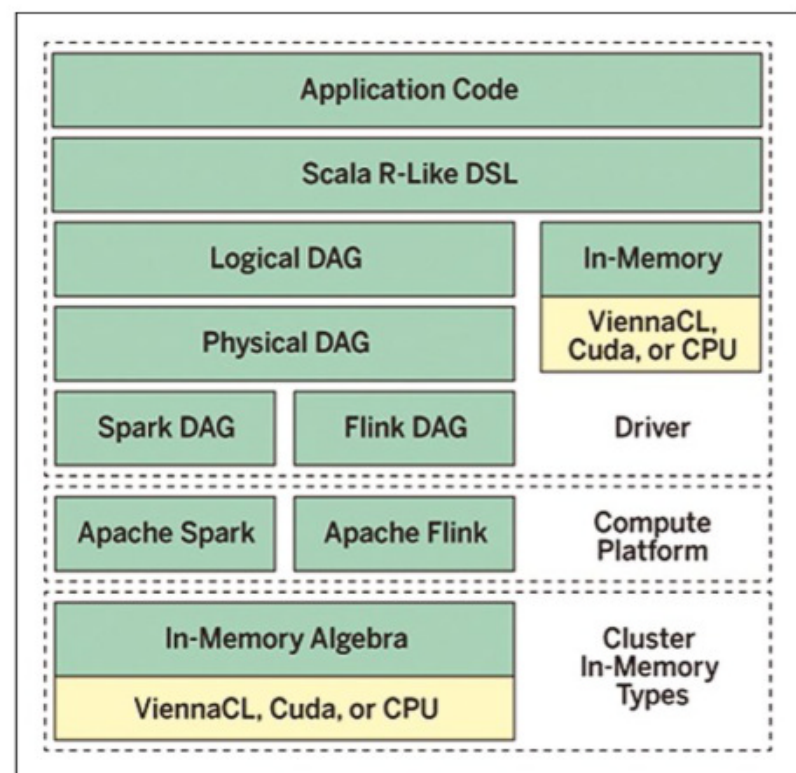


Figure 2: Architectural diagram of an engine abstraction layer.

Listing 1: Samsara Syntax Examples

```
val G = B %*% B.t - C - C.t + (xi dot xi) * (s_q cross s_q)
// Dense vectors:
val denseVec1: Vector = (1.0, 1.1, 1.2)
val denseVec2 = dvec(1, 0, 1, 1, 1, 2)

// Sparse vectors:
val sparseVec1: Vector = (5 -> 1.0) :: (10 -> 2.0) :: Nil
val sparseVec1 = svec((5 -> 1.0) :: (10 -> 2.0) :: Nil)

// Dense matrices:
val A = dense((1, 2, 3), (3, 4, 5))

// Sparse matrices:
val A = sparse(
  (1, 3) :: Nil,
  (0, 2) :: (1, 2.5) :: Nil
)

// Diagonal matrix with constant diagonal elements:
diag(3.5, 10)

// Diagonal matrix with main diagonal backed by a vector:
diagv((1, 2, 3, 4, 5))

// Identity matrix:
eye(10)

// Plus/minus:
a + b
a - b
a + 5.0
a - 5.0

// Hadamard (elementwise) product:
a * b
a * 0.5

// Operations with assignment:
a += b

a -= b
a += 5.0
a -= 5.0
a *= b
a *= 5

// Dot product:
a dot b

// Cross product:
a cross b

// Matrix multiply:
a %*% b

// Optimized right and left multiply with a diagonal matrix:
diag(5, 5) %*% b
A %*%: diag(5, 5)

// Second norm, of a vector or matrix:
a.norm

// Transpose:
val Mt = M.t

// Cholesky decomposition
val ch = chol(M)

// SVD
val (U, V, s) = svd(M)

// In-core SSVD
val (U, V, s) = ssvd(A, k = 50, p = 15, q = 1)

// EigenDecomposition
val (V, d) = eigen(M)

// QR decomposition
val (Q, R) = qr(M)
```



```
%flinkMahout

// Imports and creating the distributed context, similar but not exactly the same
import org.apache.flink.api.scala._
import org.apache.mahout.math.drm._
import org.apache.mahout.math.drm.RLikeDrmOps._
import org.apache.mahout.flinkbindings._
import org.apache.mahout.math._
import scalabindings._
import RLikeOps._

implicit val ctx = new FlinkDistributedContext(benv)

// CODE IS EXACTLY THE SAME FROM HERE ON - R-Like DSL
val drmData = drmParallelize(dense(
  (2, 2, 10.5, 10, 29.509541), // Apple Cinnamon Cheerios
  (1, 2, 12, 12, 18.042851), // Cap'n'Crunch
  (1, 1, 12, 13, 22.736446), // Cocoa Puffs
  (2, 1, 11, 13, 32.207582), // Froot Loops
  (1, 2, 12, 11, 21.871292), // Honey Graham Ohs
  (2, 1, 16, 8, 36.187559), // Wheaties Honey Gold
  (6, 2, 17, 1, 50.764999), // Cheerios
  (3, 2, 13, 7, 40.400208), // Clusters
  (3, 3, 13, 4, 45.811716)), numPartitions = 2)

drmData.collect(0 until 4)

val drmX = drmData.collect(0 until 4)
val y = drmData.collect(0 until 4)
val drmXtX = drmX.t %%% drmX
val drmXty = drmX.t %%% y

val XtX = drmXtX.collect
val Xty = drmXty.collect(0 until 4)
val beta = solve(XtX, Xty)
```

Figure 3: Matrix math with the Apache Flink engine.

of computation for the user. The advantage of this approach is that the end user, the person implementing the algorithms, doesn't have to know all of the peculiarities of a specific engine. In fact, the person writing the algorithms doesn't even need to know what the underlying engine is.

Samsara: The Scala DSL

The end user interfaces with Samsara Scala DSL, which makes it much more pleasant and natural for mathematicians and the writers of algorithms to implement mathematics with the help of an interesting Scala feature that allows users to change syntax and language rules for a particular use case. For example, the computation below, which is used in Mahout's distributed stochastic principal component analysis (dsPCA), when written with mathematics notation here,

$$G = BB^T - C - C^T + \xi^T \xi_S^T S_q$$

is written with Samsara as shown in the first line of Listing 1. The lines that follow are more examples of typical statements exercising Samsara syntax. You can find more information about Samsara [1] online.

Engine Agnosticism (and Why It Matters)

Engine agnosticism is important to many organizations, many of whom don't even realize it. In just the 2000s, clusters have moved from Hadoop to Spark, and from Spark to Kubernetes. Teams and organizations often find themselves in an uncomfortable position of being pinned to outmoded technology, rather than bringing in costly consultants to port

business-critical algorithms and methods from an old system to a new one. The Mahout project lived through the migration from Hadoop to Spark and incorporated the lessons learned into its very fabric, making it simple to port algorithms from any arbitrary platform to any other (Figures 3 and 4). Note that the code after the import statements requires no changes. Therefore, a team that uses one back-end engine is able to migrate code onto another engine, without having to change the code performing the mathematical operations and avoiding the more error-prone part of porting code from one platform to another.

Mahout Use Cases

Mahout is for organizations who have statistical methods to run on distributed datasets but want to minimize their exposure to the technical debt that arises from writing algorithms against a specific engine that may or may not have a successful future. Mahout allows organizations to switch their systems of record while having a minimal effect on their data outputs. Mahout also lets users add linear algebra concepts to data stores that have either weak or nonexistent implementations for linear algebra

```
%sparkMahout

// Imports and creating the distributed context, similar but not exactly the same
import org.apache.mahout.math._
import org.apache.mahout.math.scalabindings._
import org.apache.mahout.math.drm._
import org.apache.mahout.math.scalabindings.RLikeOps._
import org.apache.mahout.math.drm.RLikeDrmOps._
import org.apache.mahout.sparkbindings._

implicit val sdc = org.apache.mahout.sparkbindings.SparkDistributedContext = sc2sdc(sc)

// CODE IS EXACTLY THE SAME FROM HERE ON - R-Like DSL
val drmData = drmParallelize(dense(
  (2, 2, 10.5, 10, 29.509541), // Apple Cinnamon Cheerios
  (1, 2, 12, 12, 18.042851), // Cap'n'Crunch
  (1, 1, 12, 13, 22.736446), // Cocoa Puffs
  (2, 1, 11, 13, 32.207582), // Froot Loops
  (1, 2, 12, 11, 21.871292), // Honey Graham Ohs
  (2, 1, 16, 8, 36.187559), // Wheaties Honey Gold
  (6, 2, 17, 1, 50.764999), // Cheerios
  (3, 2, 13, 7, 40.400208), // Clusters
  (3, 3, 13, 4, 45.811716)), numPartitions = 2)

drmData.collect(0 until 4)

val drmX = drmData.collect(0 until 4)
val y = drmData.collect(0 until 4)
val drmXtX = drmX.t %%% drmX
val drmXty = drmX.t %%% y

val XtX = drmXtX.collect
val Xty = drmXty.collect(0 until 4)
val beta = solve(XtX, Xty)
```

Figure 4: Matrix math with the Apache Spark engine.

concepts, such as Apache Spark. (Spark's linear algebra works fine in single-node deployments but has issues scaling to larger distributed datasets.)

Mahout in the Wild

A major used car marketplace in North America used the Mahout codebase when creating their car recommendation system. This recommendation engine is based on Mahout's Correlated Cross-Occurrence (CCO) analysis. The CCO algorithm is very similar to the more popular co-occurrence (CO) algorithm, but it also incorporates other attributes of the user into its recommendations; in more technical parlance, it is multimodal. Mahout has been used in many situations where customer privacy and intellectual property concerns keep them from being published, but many

researchers and practitioners have built recommenders, similarity engines, and other predictive models at scale with the use of its tools.

A paper written by T. Grant [2] illustrates another Mahout use case. During the outset of the COVID-19 pandemic, CT scans were shown to be as good as, or in some cases superior to, RT-PCR tests. A major issue however was the high dose of radiation they delivered. With Mahout, Grant showed how "noisier," "low-dose" CT scans could be quickly and easily denoised, with about five lines of Mahout code.

Getting Started with Mahout

Mahout can be added to your project by adding it as an Apache Maven dependency, by running a prebuilt Docker image, or by downloading a binary [3] or a source build [4] from the project website. To get involved with

the project as a user or contributor, subscribe to user@mahout.apache.org and dev@mahout.apache.org mailing lists (instructions online [5]). For some alternative methods for installing and using the software, including a prebuilt Docker image, see the slide presentation at SlideShare [6]. ■

Info

- [1] Samsara: [<https://mahout.apache.org/docs/latest/mahout-samsara/faq.html>]
- [2] "Denoising COVID-19 computed tomography scans with scalable open source software," by Trevor Grant, [<https://noblereasearch.org/Content/PDF/12/2399-8172.2020-6/2399-8172.2020-6.pdf>]
- [3] Binary image: [<https://mahout.apache.org/general/downloads>]
- [4] Source build: [<https://mahout.apache.org/>]
- [5] Mailing lists: [<https://mahout.apache.org/general/mailling-lists>]
- [6] SlideShare: [<https://www.slideshare.net/AndrewMusselman/apache-mahout-on-zeppelinptx>]

Shop the Shop → shop.linuxnewmedia.com

Discover the past and invest in a new year of IT solutions at Linux New Media's online store.

Want to subscribe?

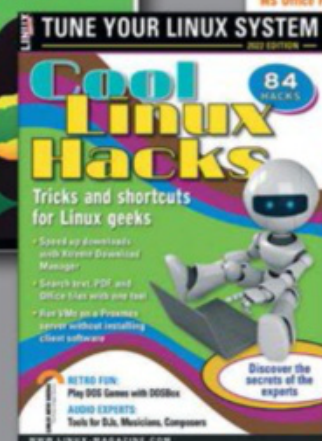
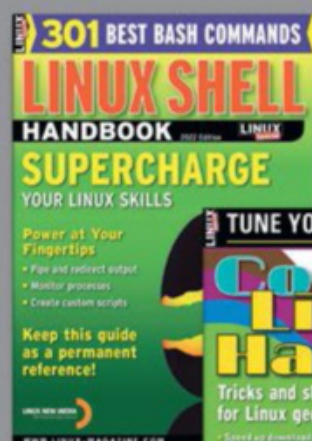
Searching for that back issue you really wish you'd picked up at the newsstand?

➤ shop.linuxnewmedia.com

DIGITAL & PRINT
SUBSCRIPTIONS



SPECIAL EDITIONS



Foundries.io IoT development platform

Conundrum Solver

Foundries.io is a modular system for companies wanting to develop Internet of Things applications for devices. By Martin Loschwitz

If you want to see die-hard system administrators go mad, all you need to do is bring up the topic of the Internet of Things (IoT). Virtually no other a topic will unite Linux admins in such a unanimous opinion. Stories offer examples of IoT applications that have gone wildly wrong. Many of these are urban legends and never actually happened, but they fit beautifully into the narrative of technology that doesn't really help anyone, opens security holes, and, in the eyes of many, simply shouldn't exist. However, valid IT approaches do exist, and IoT devices perform well every day. The fact that the nation's discounters regularly offer WiFi-capable cameras with woeful security measures should not obscure the fact that in many places around the world, networked WiFi cameras with good security can protect people. In other words, IoT applications often are not as useless as some people would make them out to be (**Figure 1**). This is where Foundries.io steps into the breach by offering Foundries Factory as a complete solution for companies that want IoT functionality but do not want to deploy the entire infrastructure in-house. The package

includes a software stack, a standardized development environment, and a pricing model that avoids per device charges.

Importance of Security

That said, the mistrust of IoT is not entirely unjustified. Undoubtedly, a washing machine with a WiFi module for remote app control that can be hijacked by attackers on the web is not a good thing. Unfortunately, precisely this kind of attack has been extensively documented in the operations security (OpSec) scene. Often the companies that regularly make the biggest security mistakes are those that have little or nothing to do with software.

Just because you build good washing machines does not automatically make you an innovative IT company. Companies looking to give their devices a modern and smart touch regularly source modules externally and install them more or less with no inkling of what might go wrong. When the appliance reaches the customer's location and turns out to have an intrinsic security problem, both the customer and manufacturer

are powerless, because most IoT devices don't even provide for updates over the network to patch security holes that have become public. The service engineers who replace

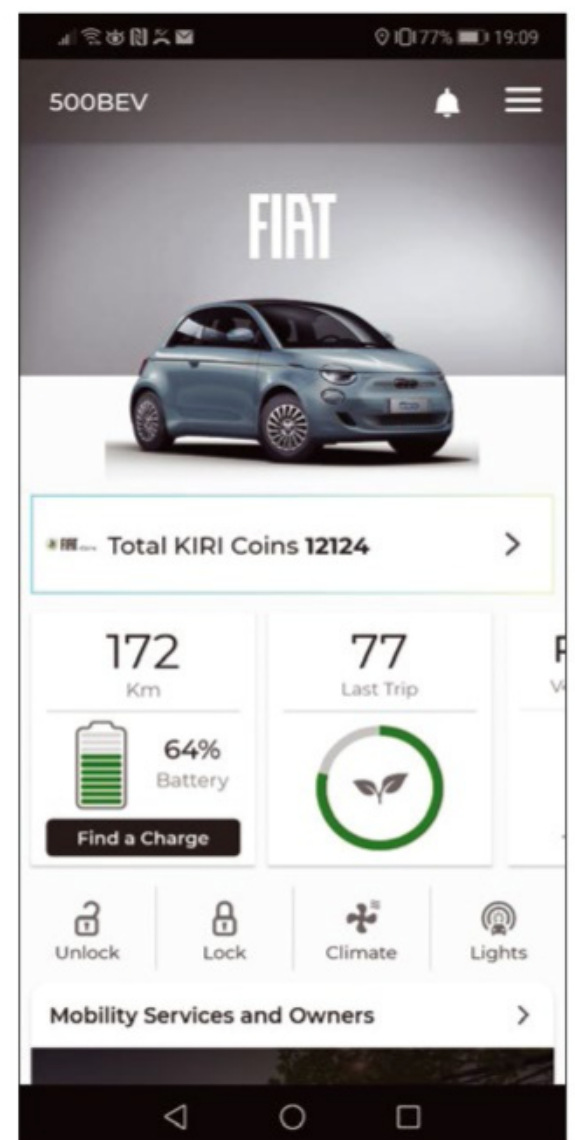


Figure 1: Today, even cars have an Internet connection and communicate over a smartphone, making them part of the Internet of Things.

Photo by Juan Rumimpunu on Unsplash

relays and bearings in washing machines are highly unlikely qualified to help customers install new firmware. Often, manufacturers looking to jump on the IoT bandwagon to reach state-of-the-art status lack a plan and have no idea how IoT can be implemented in a meaningful way.

Foundries.io

In essence, the Foundries stack is a framework into which external developers can build functionality for a device while having access in the background to the centralized skill set of a large provider, including the corresponding logistics.

If you use the Foundries stack, the marketing people promise, you no longer have to develop the major share of an IoT stack for any device yourself; rather, you can rely on a modular system that already contains the most important functions. I took an in-depth look at Foundries.io, so I could explain how the solution works, what its advantages and disadvantages are, and whether the product gives rise to hope of more secure and capable IoT applications.

What IoT Needs

What do manufacturers need to establish IoT-enabled devices successfully on the market? I have already touched on a few factors, including the inability of a company producing household appliances to develop IoT software in-house and their reliance on off-the-peg components available on the market. Unfortunately, a software stack alone does not make an IoT application; the hardware (Figure 2) also plays an important role.

A toaster that is supposed to notify the owner over WiFi that the toast is ready is a good way of proving that IoT applications require more hardware than you might at first think. A mandatory requirement is some kind of CPU, and it needs a sensor that can detect that the toast is finished. In the simplest case, the sensor might simply detect that the toaster has automatically switched itself off and

ejected the toast after the time set by the user. This would be a fairly trivial implementation.

For most IoT developers, though, this basic functionality doesn't go far enough. Ideally, you would want the toaster to stop the toasting process automatically by the level of browning set in an app, without having to specify a time. This scenario requires significantly more technology, such as a sensor that can measure the degree of browning. Another function allows the built-in computer to interrupt the toasting process and eject the toast as soon as it is ready. Also, the toaster needs a network connection. Modern devices rely exclusively on WiFi, which requires a matching chip and antenna.

The hardware bill of materials leads to the conclusion that the software used in a device of this type must be capable of far more than just a few simple network commands. The task is to read and interpret values from sensors. The operating system – because even the toaster needs one – needs drivers for the sensors and components, and you need some kind of software in userland that links the features of these components in a meaningful way.

If you don't think the toaster example is practical enough at this point, imagine a similar scenario with a smart washing machine or refrigerator. The combination of a computer board with networking capability and a variety of sensors are found in almost any scenario.

Stack and Operating System

To get back to the Foundries stack, it should first be noted that the product exclusively relates to the software part of an IoT application. Foundries.io doesn't build the hardware, but the manufacturer does maintain partnerships with contractors with operations in the IoT market.

IoT devices are almost always embedded. Accordingly, devices with an ARM system on a chip (SoC) are widespread; the entire computer comprises a small circuit board with

all the relevant components, which is exactly where the Foundries stack (Figure 3) comes in.

The innermost core is a Linux kernel that includes support for a wide range of popular ARM boards for IoT deployment. The kernel is enriched with drivers for chips that are typically used in the IoT environment, for example, (W)LAN devices. The project's website provides a list of boards [1] that can be used directly with the software stack provided by the Foundries project.

Although this solution does not sound like much in theory, it is, in practice, a massive boost for companies looking to get started with IoT devices. Thanks to the preparatory work by Foundries.io, a basic system is available within minutes on which further development can be built, provided

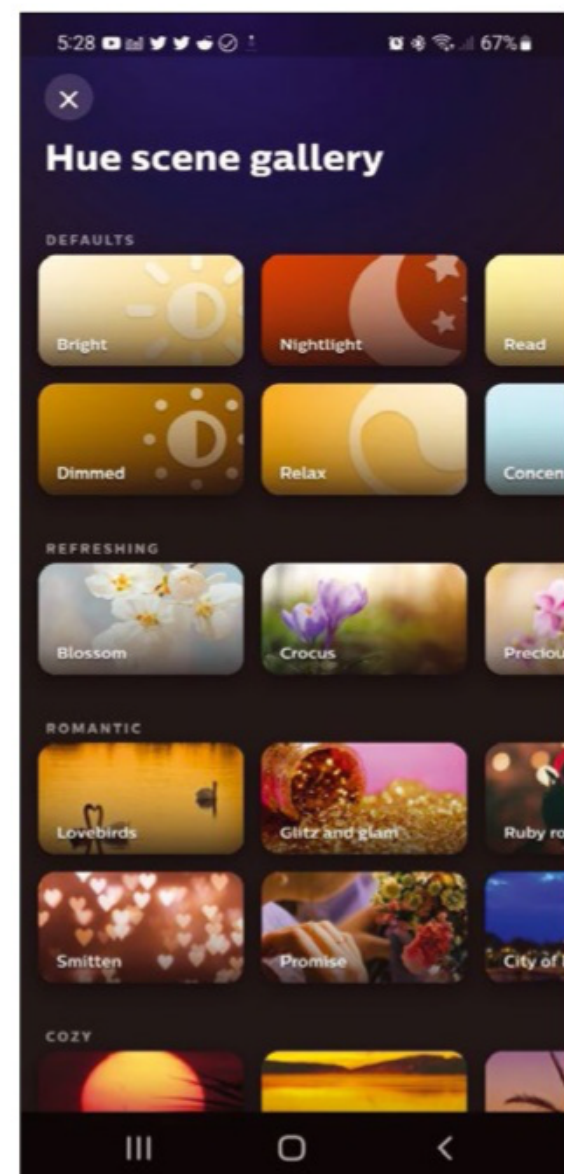


Figure 2: Every IoT device (e.g., the smart lights by Philips in this example) is basically a small computer with a system-on-a-chip board that requires an operating system. Foundries.io offers the Factory development environment to help manufacturers achieve results quickly.

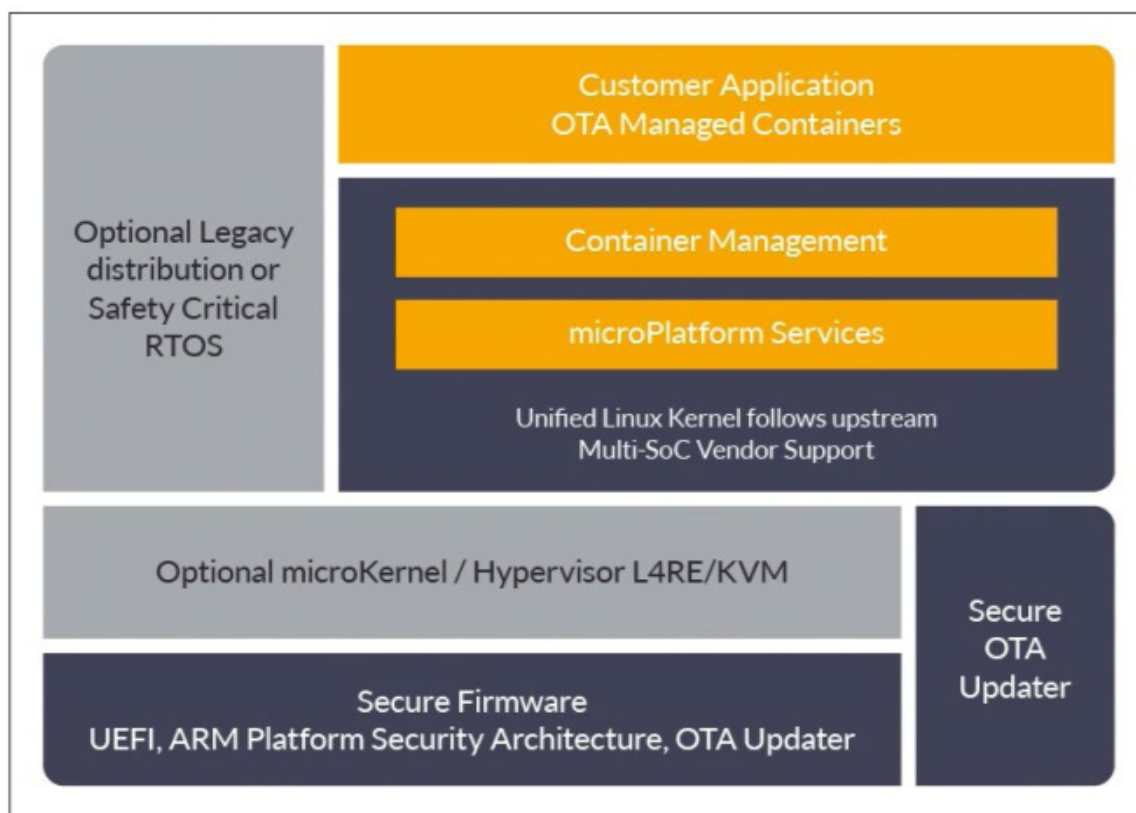


Figure 3: Foundries.io provides enterprises with a development platform for IoT applications that has a Linux kernel with a container platform and microservices at its core.
© Foundries.io

that a suitable SoC board is available. Without Foundries.io, just putting together a suitable Linux distribution for embedded devices would take a medium-sized team months.

More Than Linux

Linux, by the way, is not the only operating system with which the Foundries developers planned to work. During its startup phase, they dropped quite a few hints in the documentation and online that the company had its sights set on an embedded distribution based on the Zephyr real-time system. Zephyr, like the Linux kernel, is under the aegis of the Linux Foundation and specializes in real-time computing.

In the meantime, however, the references to Zephyr have disappeared from the vendor's documentation and website, and the Zephyr-based distribution is probably no longer maintained. However, it would only have appealed to a relatively small group of users anyway, because real-time computing is only likely to play a minor role given the typical use cases in the IoT environment. The developers are also aware that an operating system kernel is not the same as a functional IoT framework.

Security

For obvious reasons, the issue of security plays a major role for IoT devices. On one hand, these devices are not shielded from the outside world as much as you might assume. Recall once again the example of surveillance cameras: Because they support the Universal Plug and Play (UPnP) standard and many routers for domestic use are configured to pass automatically through ports released by UPnP to the outside world, the corresponding cameras suddenly become accessible on the web.

On the other hand, many owners would not even notice an attack on smart home devices, as long as the basic functions of the device are not affected. If an attacker were able to take control of the IoT toaster described above and a few thousand more devices with the same vulnerability to execute arbitrary code, it would even be possible to imagine a botnet populated just by toasters. Several approaches are available to prevent this kind of attack. For example, known vulnerabilities could be repaired by patching. Another approach would be not to allow the execution of arbitrary code on the

devices in the first place. What has been an established standard on desktop and server systems in the form of the Trusted Platform Module (TPM) for a long time also exists for embedded systems. But most providers do not make use of these options.

Foundries.io approaches the problem differently: It fully supports the security features of any hardware on which it can run. From the bootloader to individual drivers and programs, a chain of trust can be created that prevents the execution of arbitrary code, even if an attacker is working as root on the system. The Foundries stack scores bonus points because it can also natively use the cryptographic functions of many ARM and Intel chips on the market to enable efficient encryption. Therefore, developers can use encrypted connections instead of plain text, further contributing to device security.

Updates

Updates for IoT devices are a complicated affair. In practice, they can only occur "over the air" (OTA; i.e., in a way that does not entail the provider having physical access to the device after delivery). The distribution model for updates accordingly provides that they be made available on central servers so that clients can download them autonomously. In practice, of course, this also means that an IoT device needs access to the Internet at its location, which is not a problem in most cases. Most providers integrate their IoT devices into existing WiFi networks that have a connection to the Internet. The provider of the respective software has to take care of the rest.

Foundries.io is up to pace here, too. Every part of the operating system – from the bootloader to the kernel to userspace – can easily be updated remotely according to defined standards. From a security point of view, this is a smart implementation. If you implement IoT devices with Foundries.io, you can, for example,

specify that updates always need a digital signature for the target devices to import them. In this way, a provider can prevent attackers from hijacking its devices by loading a hacked firmware update. For user-space updates, Foundries.io relies on *libostree* (previously OSTree) to update individual parts right down to individual files, ensuring that updates can be installed regularly and incrementally instead of in the form of major release cycles.

Beyond IoT

The examples used in this article so far refer to devices from the IoT world, which are more likely to be found in a domestic environment. However, IoT now also plays a major role in industry, and many functions for this purpose can also be found in the Foundries stack. If you want to roll out IoT devices as edge applications in the enterprise, for example, you might need an option to execute commands directly from a safe distance. The developers have implemented this scenario by including the WireGuard virtual private network (VPN) client. If configured appropriately, systems with the Foundries stack open a VPN connection to their home address and are then open to receiving instructions. The native cloud connectors for the major hyperscalers that come with the product are also aimed at business customers. For example, services and instances on AWS or Azure can be started or manipulated from the system without you having to implement the interface to these cloud environments yourself. Again, Foundries.io saves you considerable overhead by including a ready-made solution for a standard task. The same applies to support for the protocols typically found in the IoT environment, which is included by default. For example, if the device has a chip with Zigbee support (Zigbee being the protocol for controlling smart lighting), Foundries.io provides an interface for it in the system. The same applies to the Open Mobile

Alliance Lightweight Machine-to-Machine (OMA LwM2M) protocol, which specifies several standard communication methods between IoT devices, such as message queuing telemetry transport (MQTT) over HTTP. The following basically applies: Once a protocol has established itself as a standard in the IoT environment, the chances are good that the Foundries stack will support it.

Enter Docker

The Foundries stack comes with Docker as the runtime engine for containers, and third-party vendors are advised to integrate their specific customizations into the Foundries framework in the form of Docker containers for several reasons. Like the rest of the system, Docker containers in the Foundries stack can be upgraded to new versions with OTA updates. Therefore, vendors can effectively eliminate errors in IoT devices – even in the vendor-specific software components. Moreover, companies benefit from clear demarcation between the operating system on the one hand and their own applications on the

other. Decoupling is also useful from a development point of view, as well as in everyday operation, because it enables far more granular work than monolithic firmware.

Additional Functions

Up to this point, my main focus has been on the Foundries stack (i.e., the distribution for embedded systems that forms the core of the Foundries.io portfolio). However, the provider's service is not limited merely to offering a ready-for-use image of this distribution. Instead, Foundries Factory presents itself as a full-blown development environment that can be used to produce versions of the embedded distribution that are highly adapted to individual devices.

The first step for companies that want to use Foundries Factory is therefore likely to be creating an account on the provider's website. Doing so creates an entire development environment in which the Foundries stack is already present as a core component. Of course, none of this just falls into place accidentally. The Foundries developers primarily rely on Yocto [21]; in turn, Yocto dynamically generates

Figure 4: Foundries Factory offers comprehensive fleet management. Embedding a corresponding function in your firmware means you can ensure that remote devices automatically register with Factory as soon as they have an Internet connection. © Foundries.io

a development environment for embedded devices on the basis of the OpenEmbedded project.

In this environment, companies then have the ability to customize the vendor's generic embedded distribution to suit the IoT device of their dreams. Foundries Factory delivers bootable images at the push of a button, and the images can be forwarded to a company's hardware manufacturer, who can then install the software directly on the devices. The vendor supports developers with, for example, native Git integration in Foundries Factory or by providing a complete continuous integration/continuous deployment (CI/CD) build chain. Once the devices reach the customer's data center or living room, Foundries Factory offers comprehensive fleet management (Figure 4), which is more or less the server side of the OTA update process: When the vendor of a device provides an updated image for a single component of the system from Foundries Factory,

the target systems download it gradually while the Internet connection is up and running.

If you want to sell your customers remote maintenance and management in addition to IoT devices, Foundries Factory again has all the necessary components. Management capabilities are an essential part of the offering, precisely because they efficiently enable long-term management of delivered devices (Figure 5).

Not Cheap, But Inexpensive

The Foundries.io pricing model differs from the usual approach to such solutions in one important respect: Foundries.io charges flat rates and does not seek to make money on every supported device. There is also no such thing as a complete product structure with different editions. Instead, you have exactly one option – the Enterprise Factory package, which costs \$5,000 per month or \$50,000 per year and includes all the

described functions for an unlimited number of devices and unlimited numbers of builds in the Foundries Factory CI/CD environment. Of course, this also means that the more devices a provider sells with this software, the more sense it makes to use the solution.

Conclusions

Foundries Factory proves that IoT devices do not have to become obsolete shortly after delivery and therefore do not have to pose a security risk.

In particular, the product is aimed at companies that want to open up IoT options for their devices virtually from scratch, without having to develop a basic system themselves. Foundries.io provides a complete toolbox from which suitable applications can be easily assembled. In addition to a full-blown development environment, it contains a ready-made mini-distribution based on the Linux kernel with a runtime environment for Docker containers. These tools can significantly reduce a manufacturer's time to market for an IoT device. Commercial use of Foundries Factory should be well worth their while in most cases.

Technically, the solution is cutting edge. Extensive support for a large number of security functions is just as useful as the cleverly thought-out update process, which allows all the components of an entire fleet of IoT devices to be replaced individually. Against this background, the manufacturer's prices might not look cheap, but they are inexpensive in relation to performance, especially when you consider that Foundries.io, unlike other manufacturers, offers fixed prices and does not look to earn money on every unit of a device that is sold.

Anyone who needs a powerful environment for developing and running IoT applications definitely needs to check out Foundries.io. Happily, it is quite simple to try Foundries out: The test account is available free online, and because Foundries Factory offers support for a variety of SoC boards with an ARM or Intel CPU, a single Raspberry Pi is all you need to test the deployment on hardware. ■

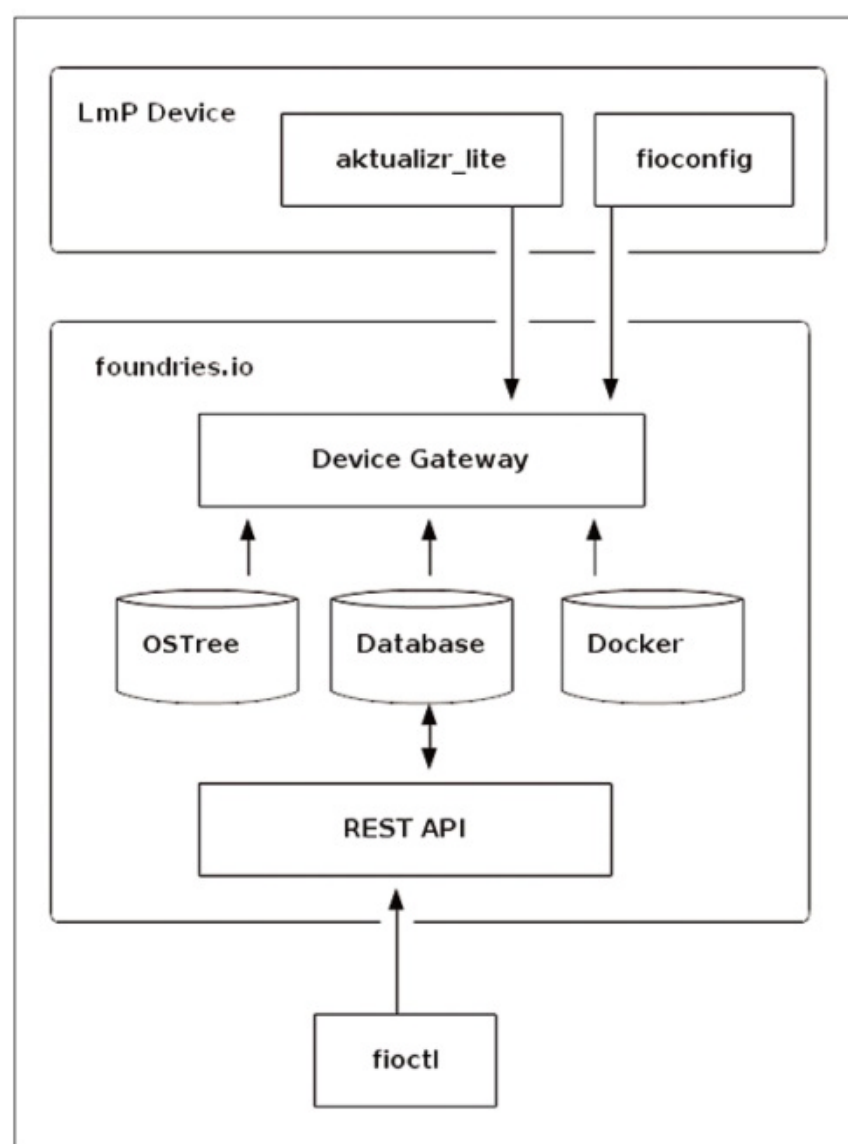


Figure 5: OTA updates are essential for IoT devices. The Foundries stack provides the ability to replace components with a newer version at any level of its architecture. © Foundries.io

Info

- [1] Supported boards: [\[https://docs.foundries.io/latest/reference-manual/linux/linux-supported.html\]](https://docs.foundries.io/latest/reference-manual/linux/linux-supported.html)
- [2] Yocto: [\[https://www.yoctoproject.org\]](https://www.yoctoproject.org)

The author

Freelance journalist Martin Gerhard Loschwitz focuses primarily on topics such as OpenStack, Kubernetes, and Chef.

Register by October 14 for Big Discounts!



You. Dallas. SC22.

NOVEMBER 13–18

One of the world's largest gatherings of HPC researchers and professionals, SC22 is an incredible week of learning new skills, forging new friendships, solving complex problems, and viewing next-gen technology.

A new, custom-built Digital Experience is available if you prefer to attend virtually.



The International Conference for High Performance
Computing, Networking, Storage, and Analysis

Register Today!

sc22.supercomputing.org



SC22

Dallas, TX | hpc
accelerates.

Sponsored by:



IEEE
COMPUTER
SOCIETY



sig hpc

PowerDNS Authoritative server high availability with MariaDB Galera

Power Up

Combining the PowerDNS Authoritative server daemon with MariaDB's multiprimary Galera cluster allows a simple yet robust solution for your DNS needs. By Donnie Greer

Recently, I found myself in the need for a trio of Authoritative nameservers to disperse between my company's data centers. Having used a PowerDNS Recursive server for years, I was anxious to give their Authoritative version a heaping helping of DNS records.

Unlike PowerDNS Recursive, the Authoritative server requires a back-end system to store records. The list of supported back ends is rather lengthy, including but not limited to MySQL, PostgreSQL, Berkeley Internet Name Domain (BIND), and even Lightweight Directory Access Protocol (LDAP). I consider myself rather skilled at MariaDB, and because the Authoritative server supports MySQL, I knew that MariaDB would be a non-issue.

I've set up dozens of MySQL replication servers over the years, but I wanted to investigate something different, something a bit better suited to this project's specific needs. MySQL Replication uses a primary server to update one or more replicas, and because the transactions are committed sequentially, a slow transaction can cause replicas to trail behind the primary server. If the primary

fails, it is entirely possible that the replica might not have recorded the last few transactions. With a transaction-safe engine, such as InnoDB, a transaction will either be completed on replica nodes or not at all. That just won't do.

Enter MariaDB's Galera Cluster.

Galera is a virtually synchronous multiprimary cluster for MariaDB that is only available on Linux and only supports the InnoDB engine for storage (although MyISAM and Aria are in the works). With Galera, you get virtually synchronous replication, active-active multiprimary topology, read/write to any node, automatic membership control, automatic node joining, true parallel replication, and direct client connections. Those features translate to no replica lag, no lost transactions, read scalability, and smaller client latencies – perfect for keeping DNS records happy and healthy across data centers.

Installation

PowerDNS installation and setup has been covered at length in previous articles [1] [2]. I highly recommend

both articles to get your Authoritative server into a solid configuration. Assuming PowerDNS is installed and configured with the MySQL back end, you should crunch away on installing MariaDB. The first step is to download and run the MariaDB repo setup script:

```
wget https://downloads.mariadb.com/
MariaDB/mariadb_repo_setup
chmod +x mariadb_repo_setup
sudo ./mariadb_repo_setup --mariadb-server-version="mariadb-10.5"
```

MariaDB documentation recommends installing dependencies separately to avoid conflicting packages from your OS vendor:

```
sudo dnf install perl-DBI libaio
libsepol libselinux boost-program-options
sudo dnf install
--repo="mariadb-main" MariaDB-server
```

Once dependencies have been addressed, run the installation command for MariaDB and start the MariaDB server with systemctl:

```
sudo mysql_install_db
sudo systemctl start mariadb.service
sudo mysql_secure_installation
```


The final line ensures the use of basic security best practices.

Creating and Populating

A quick edit of your pristine MariaDB server's configuration file `/etc/my.cnf.d/server.cnf` (1) binds the MariaDB service to the localhost and (2) connects and (3) creates the database:

```
Bind-address = 127.0.0.1
mysql -h localhost -u root -p
create database pdns;
```

Listing 1 shows a quick user addition to the database and the granting of permissions. To exit from the MariaDB shell, use the `quit;` command. For PowerDNS to work as intended, you need to add the default schema (**Listing 2** for PowerDNS 4.3). Schemas for PowerDNS version 4.2 or 4.1 can be found on the PowerDNS documentation website [\[3\]](#) [\[4\]](#).

The basic configuration is complete, but you have a bit more to do. MariaDB needs to know that you intend to use Galera to cluster your PowerDNS database. To do that, edit `/etc/my.cnf.d/server.cnf` and add or modify the variables shown in **Listing 3**. Pay special attention to the `wsrep_cluster_address` variable because it is the list of IP addresses of all nodes in the cluster. To

Listing 1: User Creation

```
GRANT ALL ON pdns.* TO 'pdnsadmin'@'localhost' IDENTIFIED BY 'CreateAnAwesomePassword';
GRANT ALL ON pdns.* TO 'pdnsadmin'@'localhost.localdomain' IDENTIFIED BY 'CreateAnAwesomePassword';
FLUSH PRIVILEGES;
```

Listing 2: Populate a Database

```
01 use pdns;
02
03 CREATE TABLE domains (
04   id                INT AUTO_INCREMENT,
05   name              VARCHAR(255) NOT NULL,
06   master            VARCHAR(128) DEFAULT NULL,
07   last_check        INT DEFAULT NULL,
08   type              VARCHAR(6) NOT NULL,
09   notified_serial    INT UNSIGNED DEFAULT NULL,
10   account            VARCHAR(40) CHARACTER SET 'utf8' DEFAULT NULL,
11   PRIMARY KEY (id)
12 ) Engine=InnoDB CHARACTER SET 'latin1';
13
14 CREATE UNIQUE INDEX name_index ON domains(name);
15
16 CREATE TABLE records (
17   id                BIGINT AUTO_INCREMENT,
18   domain_id          INT DEFAULT NULL,
19   name              VARCHAR(255) DEFAULT NULL,
20   type              VARCHAR(10) DEFAULT NULL,
21   content            VARCHAR(64000) DEFAULT NULL,
22   ttl                INT DEFAULT NULL,
23   prio              INT DEFAULT NULL,
24   disabled           TINYINT(1) DEFAULT 0,
25   ordername          VARCHAR(255) BINARY DEFAULT NULL,
26   auth              TINYINT(1) DEFAULT 1,
27   PRIMARY KEY (id)
28 ) Engine=InnoDB CHARACTER SET 'latin1';
29
30 CREATE INDEX nametype_index ON records(name,type);
31 CREATE INDEX domain_id ON records(domain_id);
32 CREATE INDEX ordername ON records (ordername);
33
34 CREATE TABLE supermasters (
35   ip                VARCHAR(64) NOT NULL,
36   nameserver         VARCHAR(255) NOT NULL,
37   account            VARCHAR(40) CHARACTER SET 'utf8' NOT NULL,
38   PRIMARY KEY (ip, nameserver)
39 ) Engine=InnoDB CHARACTER SET 'latin1';
40 CREATE TABLE comments (
41   id                INT AUTO_INCREMENT,
42   domain_id          INT NOT NULL,
43   name              VARCHAR(255) NOT NULL,
44   type              VARCHAR(10) NOT NULL,
45   modified_at        INT NOT NULL,
46   account            VARCHAR(40) CHARACTER SET 'utf8' DEFAULT NULL,
47   comment            TEXT CHARACTER SET 'utf8' NOT NULL,
48   PRIMARY KEY (id)
49 ) Engine=InnoDB CHARACTER SET 'latin1';
50
51 CREATE INDEX comments_name_type_idx ON comments (name, type);
52 CREATE INDEX comments_order_idx ON comments (domain_id, modified_at);
53
54 CREATE TABLE domainmetadata (
55   id                INT AUTO_INCREMENT,
56   domain_id          INT NOT NULL,
57   kind              VARCHAR(32),
58   content            TEXT,
59   PRIMARY KEY (id)
60 ) Engine=InnoDB CHARACTER SET 'latin1';
61
62 CREATE INDEX domainmetadata_idx ON domainmetadata (domain_id, kind);
63
64 CREATE TABLE cryptokeys (
65   id                INT AUTO_INCREMENT,
66   domain_id          INT NOT NULL,
67   flags              INT NOT NULL,
68   active             BOOL,
69   published          BOOL DEFAULT 1,
70   content            TEXT,
71   PRIMARY KEY (id)
72 ) Engine=InnoDB CHARACTER SET 'latin1';
73
74 CREATE INDEX domainidindex ON cryptokeys(domain_id);
75
76 CREATE TABLE tsigkeys (
77   id                INT AUTO_INCREMENT,
78   name              VARCHAR(255),
79   algorithm          VARCHAR(50),
80   secret            VARCHAR(255),
81   PRIMARY KEY (id)
82 ) Engine=InnoDB CHARACTER SET 'latin1';
83
84 CREATE UNIQUE INDEX namealgoindex ON tsigkeys(name, algorithm);
```


Listing 3: Galera Config

```
[galera]
wsrep_on=ON
wsrep_cluster_name=MyHappyLittleCluster
wsrep_provider=/usr/lib64/galera-4/libgalera_smm.so
wsrep_cluster_address=gc
omm://192.168.0.10,192.168.0.20,192.168.0.30
binlog_format=row
innodb_autoinc_lock_mode=2
```

add more nodes, simply add their IP address separated by a comma. Take note that the variables shown in [Listing 3](#) are the minimal, mandatory variables to make Galera a happy camper, but many more tunable goodies can be found in Galera Variables documentation [\[5\]](#). Additionally, the new cluster nodes will attempt to connect to other nodes listed in `wsrep_cluster_address` in search of a Primary Component, which will be the first node you bootstrap with the `galera_new_cluster` script. To bootstrap a new cluster and create a new Primary Component, run the command

```
sudo galera_new_cluster
```

only on the first node. This command identifies the first node as a “seed” to populate the databases of newly added nodes. Therefore, all nodes added to the cluster will automatically copy the complete schema and data without user intervention. How cool is that?!

More Power

Once the initial node is set up, simply follow the steps above two more times to create secondary and tertiary nodes. Remember to bootstrap *only the first node* in your cluster and verify that `/etc/my.cnf.d/server.cnf` is identical on all three. The `systemctl` command starts the other nodes after the Primary Component node.

Testing

After adding more nodes to the cluster, a quick and easy test is to connect to any node in the cluster with the MariaDB client and run a quick SQL statement:

```
sudo mariadb
SHOW GLOBAL STATUS
LIKE 'wsrep_cluster_size';
```

If the size of the cluster is equal to the number of nodes in the cluster, it’s time to celebrate! If the size is smaller than expected, either a node did not start correctly, or it cannot connect to the Primary Component.

Conclusion

If you need basic and easy data replication with minimal setup time, MariaDB’s Galera Cluster has you covered. Just think what you could do with a slightly more complex configuration

with HAProxy [\[6\]](#) and Keepalived [\[7\]](#). Imagine having a single web interface that could update all PowerDNS Authoritative servers at once, thanks to Galera [\[8\]](#). ■

Info

- [1]** “Speed up Your Name Server with a MySQL Back End” by Joseph Guarino: [\[https://www.admin-magazine.com/Articles/Speed-up-Your-Name-Server-with-a-MySQL-Back-End/\]](https://www.admin-magazine.com/Articles/Speed-up-Your-Name-Server-with-a-MySQL-Back-End/)
- [2]** “Exploring PowerDNS” by Joseph Guarino: [\[https://www.admin-magazine.com/Articles/PowerDNS-The-Other-Open-Source-Name-Server/\]](https://www.admin-magazine.com/Articles/PowerDNS-The-Other-Open-Source-Name-Server/)
- [3]** PowerDNS 4.2 schema: [\[https://github.com/PowerDNS/pdns/blob/rel/auth-4.2.x/modules/gmysqlbackend/schema.mysql.sql\]](https://github.com/PowerDNS/pdns/blob/rel/auth-4.2.x/modules/gmysqlbackend/schema.mysql.sql)
- [4]** PowerDNS 4.1 schema: [\[https://github.com/PowerDNS/pdns/blob/rel/auth-4.1.x/modules/gmysqlbackend/schema.mysql.sql\]](https://github.com/PowerDNS/pdns/blob/rel/auth-4.1.x/modules/gmysqlbackend/schema.mysql.sql)
- [5]** MariaDB system variables and options: [\[https://mariadb.com/docs/deploy/community-cluster-cs10-5-rhel8/\]](https://mariadb.com/docs/deploy/community-cluster-cs10-5-rhel8/)
- [6]** HAProxy: [\[http://www.haproxy.org/\]](http://www.haproxy.org/)
- [7]** Keepalived: [\[https://github.com/acassen/keepalived\]](https://github.com/acassen/keepalived)
- [8]** PowerDNS-Admin: [\[https://github.com/PowerDNS-Admin/PowerDNS-Admin\]](https://github.com/PowerDNS-Admin/PowerDNS-Admin)

The Author

Donnie Greer is a 26-year IT veteran specializing in Linux, IaaS, and as little work as possible. He can often be heard yelling at his Synology NAS: “Work faster! I believe in you!”

REAL SOLUTIONS *for* REAL NETWORKS

ADMIN is your source for technical solutions to real-world problems.

Improve your admin skills with practical articles on:

- Security
- Cloud computing
- DevOps
- HPC
- Storage and more!

**GET IT
FAST**

with a digital subscription!

6 issues per year!

..... **ORDER NOW**

shop.linuxnewmedia.com

Certificate management with FreeIPA and Dogtag

Show Your ID

The Dogtag certificate manager integrated into the FreeIPA open source toolset generates SSL/TLS certificates for intranet services and publishes them on the network. By Andreas Stolzenberger

Both internal and external services rely on encrypted communication with SSL and TLS. For external services, administrators use officially signed certificates, although Let's Encrypt is absolutely fine in many scenarios. In contrast, internal services predominantly rely on self-signed certificates, which always cause a stir with web browsers on the local area network (LAN) by generating messages such as *The server's certificate is unknown*.

Administrators would prefer to see a nice lock icon displayed in the browser for a trusted TLS connection – for their intranet applications, too – instead of requiring users to create an individual exception in the browser for every internal application. This also means that stricter security policies can be applied for browsers on the corporate network, preventing users from opening untrusted connections at all or from creating exceptions. Admins also want other internal services to use trusted certificates and SSL for communication.

All you need is your own certificate authority (CA) on your intranet to manage and sign certificates for the connected services. Internal computers then only need to trust that this internal root CA for all keys signed by it are identified as valid.

Dogtag [1], the open source certificate system, offers a simple approach to managing an internal CA, and it integrates seamlessly with the FreeIPA [2] user directory. FreeIPA is to Linux what Active Directory (AD) is to the Windows world. It uses the same technology with a Lightweight Directory Access Protocol (LDAP) back end and Kerberos authentication. AD and an Identity, Policy, and Audit (IPA) system can trust each other with cross-domain trusts, allowing administrators of heterogeneous networks to run a connected directory for Windows and Linux machines.

In this article, I review the basic features of FreeIPA and focus on Dogtag. Every administrator of an environment with multiple Linux servers will probably run an IPA directory for central user, host, and service

management anyway, just as admins run AD on Windows networks.

Installing the FreeIPA Server

To set up the IPA server, you will want to use a Linux distribution such as Enterprise Linux distribution version 8 (EL8), Rocky, or AlmaLinux. The Minimal installation type is quite sufficient. The setup does not need special repositories such as Extra Packages for Enterprise Linux (EPEL) because all of the required packages can be found in the AppStream repository of every EL8 distribution. The IPA server must have a static IP address. Its Domain Name System (DNS) name must be configured correctly, and DNS resolution on the LAN must be correct (forward and reverse lookup). The time zone configuration must be appropriate, and the server must use Chrony to sync the time with timeservers on the web. The FreeIPA server is registered as a module named *idm* (identity

management) in the AppStream repository. The versions you will find are simply the server and the client; the dandified Yum (dnf) command lets you enable the server module and set up the required packages:

```
dnf module enable idm:DL1
dnf distro-sync
dnf module install idm:DL1/server
```

The installation is an interactive process with the

```
ipa-server-install
```

command. Alternatively, you can pass all the required parameters into the command at the command line. At this point, you need to decide whether the IPA server itself will assume the role of DNS server, which is common practice with AD. In this example, however, I assume that a DNS installation is already in place on your network.

To make it easier for IPA clients and servers to communicate, you need to add a few entries to the existing DNS server that point to the IPA installation. In this example, the server is named *ipa.mykier.ip* and it manages the *mykier.ip* realm. Running *dnsmasq* reveals the matching entries in the DNS server (Listing 1).

A Berkeley Internet Name Domain (BIND) 9 DNS server needs the same entries, but in a different format:

```
_kerberos._udp.mykier.ip. 86400 2
IN SRV 0 100 88 ipa.mykier.ip.
[...]
```

The important thing here is that both AD and IPA require the same service (SRV) records for Kerberos in DNS. If you are already using AD on the LAN, do not reassign the entries; otherwise, AD will no longer work. FreeIPA can manage without a DNS configuration. However, you then always need to include the reference to the realm and the IPA server in all operations on the clients. To enable the clients to communicate with the IPA server, you also need to open the required firewall ports on the IPA machine:

```
firewall-cmd --add-service={2
http,https,dns,ntp,freeipa-ldap,2
freeipa-ldaps} --permanent
firewall-cmd --reload
```

You can now access the web user interface (UI) on the FreeIPA server at *https://ipa.mykier.ip*. When you get there, you will see the users and hosts in your domain along with any

Listing 1: dnsmasq Output

```
srv-host =_kerberos._udp.mykier.ip,ipa.mykier.ip,88
srv-host =_kerberos._tcp.mykier.ip,ipa.mykier.ip,88
srv-host =_kerberos-master._tcp.mykier.ip,ipa.mykier.ip,88
srv-host =_kerberos-master._udp.mykier.ip,ipa.mykier.ip,88
srv-host =_kpasswd._tcp.mykier.ip,ipa.mykier.ip,88
srv-host =_kpasswd._udp.mykier.ip,ipa.mykier.ip,88
srv-host =_ldap._tcp.mykier.ip,ipa.mykier.ip,389
```

certificates issued (Figure 1). For further tests with the directory, you can now create users and groups. With the use of suitable rule sets (e.g., Sudo rules), you can allow individual users or groups to work as root users on selected hosts.

To add an existing Linux server to the directory, you first need to install the IPA client on the server. As for the IPA server, you will find the packages you need in the AppStream repository:

```
dnf install @idm:client
```

Once you have configured the IPA entries on your DNS server, a simple command is all it takes to register the server in the domain:

```
ipa-client-install --mkhomedir
```

The installer relies on DNS to discover all the information it needs

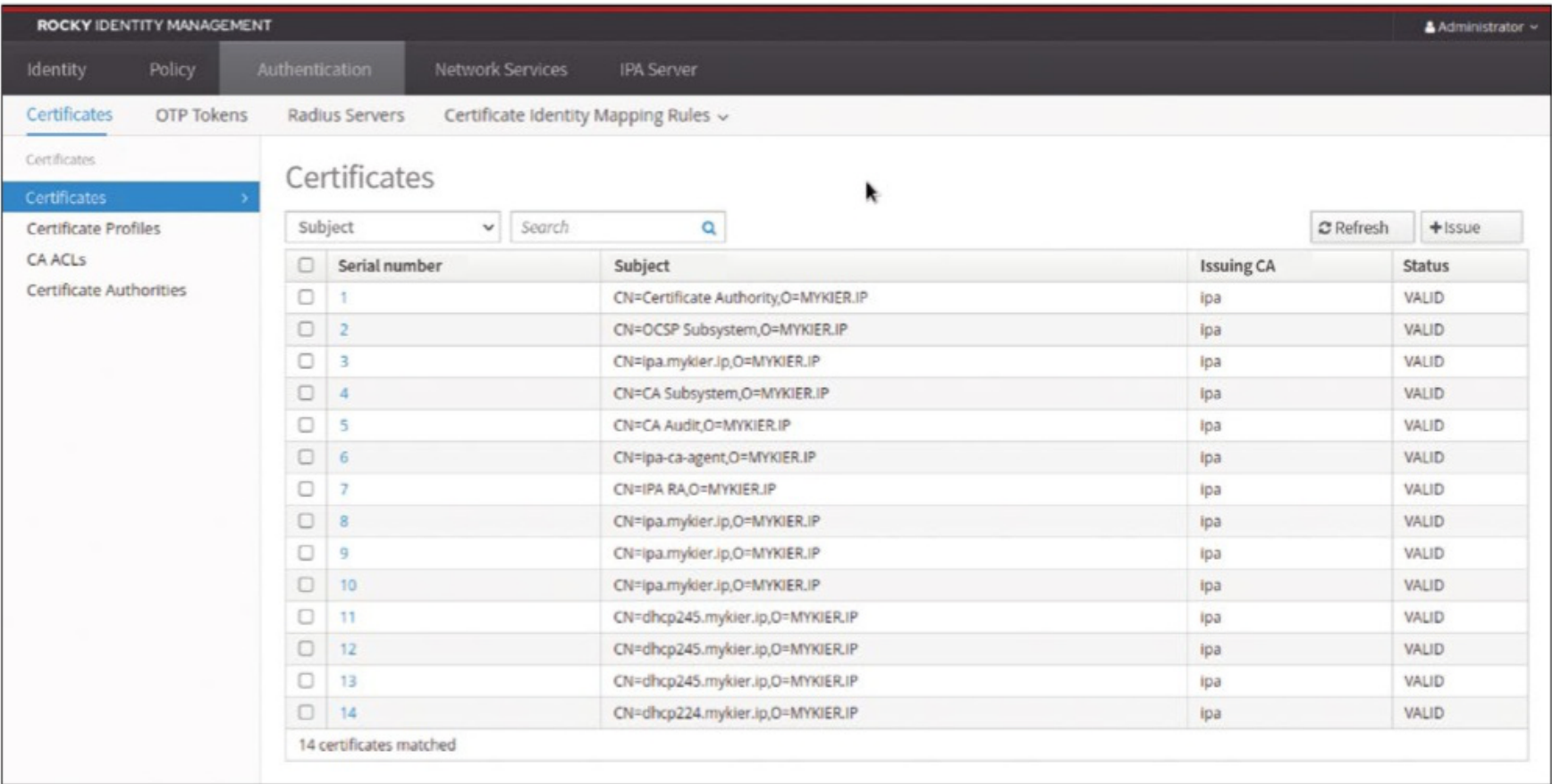


Figure 1: FreeIPA listing all the issued certificates.

about the domain. The `--mkhomedir` option tells the client to create the required home directories dynamically for the domain users on the local system. In practice, however, administrators are more likely to create an automount rule on the FreeIPA server that mounts the home directories from a central NFS server at user login time. In this way, domain users will always have access to their files, regardless from which PC within the domain they log in. Without DNS records, users need to pass in the IPA server and domain information to the command:

```
ipa-client-install 🔧
--mkhomedir 🔧
--server=ipa.mykier.ip 🔧
--domain=mykier.ip 🔧
--realm=MYKIER.IP
```

In both cases, the client setup then asks for a domain admin user account name and password to log the current machine into the directory service.

If you use Kickstart to install EL8 servers or clients, this step can be automated. If you are setting up Fedora Linux or EL8 Linux with a graphical UI (GUI) from a Live operating system disc, you can handle the domain joining step directly in the Gnome configuration after completing the basic installation. When Gnome prompts the user to create a new user, the *Enterprise Login* button pops up, which then guides you interactively through the IPA client setup.

Automating the IPA Client Rollout

When you set up a FreeIPA directory on an existing LAN, you obviously don't want to have to register manually all existing or new machines with the directory. The process can be automated in a relatively simple way. First, create the appropriate host entries in the directory service, which you can do on a computer that has the IPA client and is

authorized with admin rights to the directory:

```
ipa host-add hostname.fqdn 🔧
--password=<one-time-password>
```

The assigned password expires after the computer's first attempt to join the domain. On existing Linux hosts, install the IPA client Ansible or some other automation tool. Then, on the respective client, by remote execution (or an Ansible role), register with the domain:

```
ipa-client-install 🔧
--hostname=client.mykier.ip 🔧
--domain=mykier.ip 🔧
--mkhomedir -w <one-time-password> 🔧
--realm=MYKIER.IP 🔧
--server=ipa.mykier.ip
```

You only need the details (e.g., domain, server, and realm) if you do not have appropriate DNS records for the IPA server. By the same principle, you can automatically add systems installed by Kickstart to the directory. First define the computer, including the one-time password in the directory. In the Kickstart file, add the following options:

```
%packages
[...]
ipa-client
```

and toward the end:

```
%post
[...]
/usr/sbin/ipa-client-install 🔧
<parameters as above>
```

The system then joins the domain at the end of the automated installation process. In the web UI of your IPA server, the newly registered server is now displayed in *Identity | Hosts*. With *Policy | Sudo | Sudo rules* you could now, for example, allow one of your directory users to run the `sudo` command on the new server and perform actions as root. The ruleset also supports granular breakdown if you only

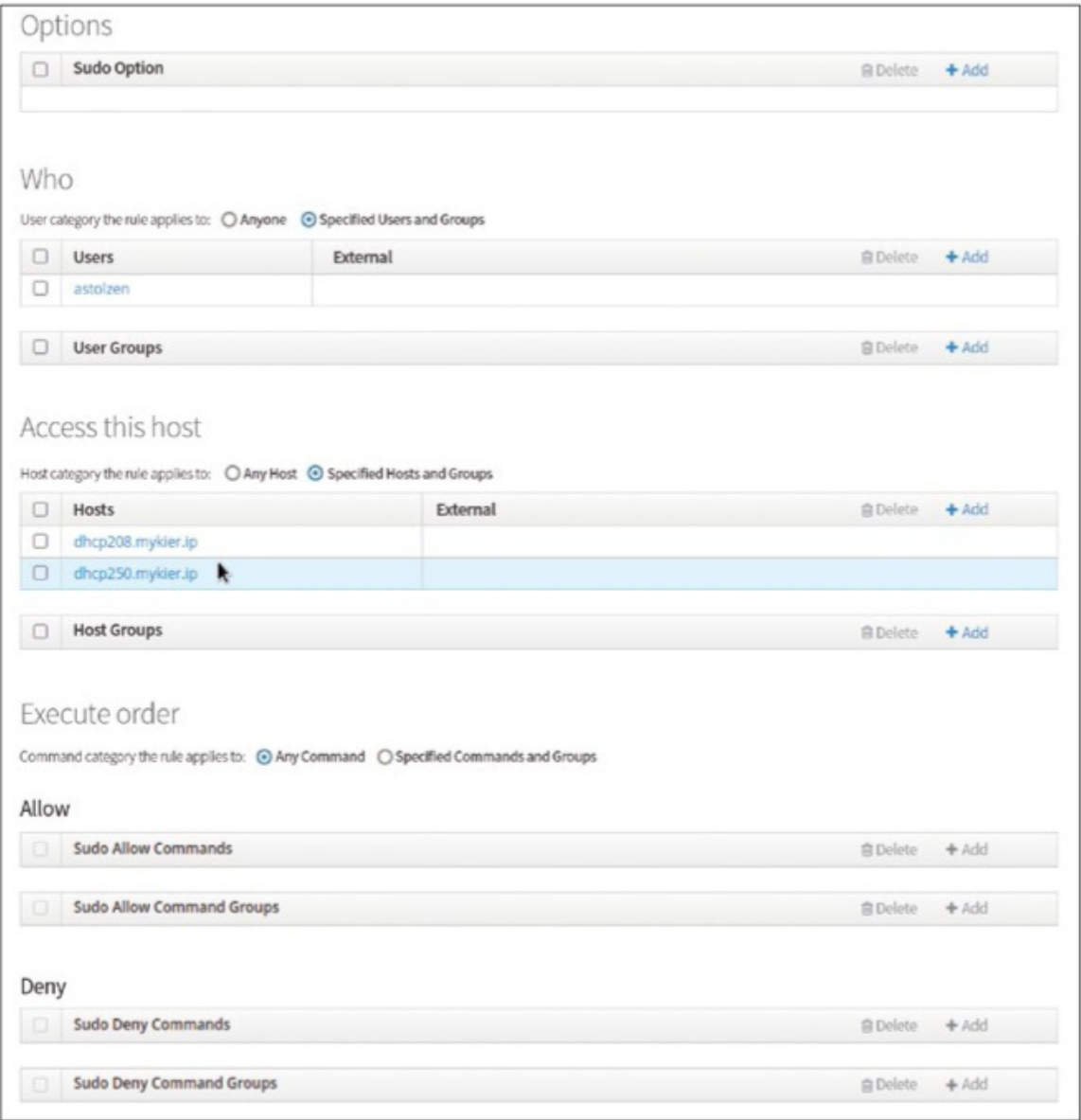


Figure 2: Sudo rules specify with which directory users can assume the root role on the various target systems. The rules can also be limited to selected commands.

want to let users run specific commands or command groups as Sudo without gaining full root access to the system (**Figure 2**).

Generating Certificates for Services

Once the new server has joined the domain, you can create certificates for services on this machine and have them signed by the root CA on the Free IPA server (**Figure 3**). In this example, I run a web server with Nginx on a freshly set up AlmaLinux 8 server that I registered in the domain as *dhcp224*. I want to give this server an official certificate for the HTTPS protocol:

```
dnf install nginx
```

Now create a suitable directory for the certificate

```
mkdir -p /etc/nginx/cert
```

EL8 systems run SELinux by default. The directory needs the correct context for the certificates; otherwise, the IPA client cannot store certificates there:

```
semanage fcontext -a 2
-t cert_t "/etc/nginx/cert(/.*)?"
restorecon -v /etc/nginx/cert
```

In the next step, log in to the directory with an admin account (*kinit admin*). Now go to the certificate directory, register the service, and request the appropriate certificate. For this case study, assume that the realm for your domain is *MYKIER.IP*:

```
cd /etc/nginx/cert
fqdn=$(hostname -f)
ipa service-add HTTP/$fqdn
ipa-getcert request -f $fqdn.crt 2
-k $fqdn.key -r 2
-K HTTP/$fqdn@MYKIER.IP -N $fqdn
```

To avoid typing the complete fully

qualified domain name (FQDN) of the server multiple times, you can define it as a shell variable named *\$fqdn*. The

```
ipa service-add
```

command registers the desired service for the server against the directory.

A number of services are predefined (e.g., *HTTP*), but you can create your own services at any time (e.g., *REDIS/redis.mykier.ip*).

The command

```
ipa-getcert request
```

creates the request and also immediately intercepts the response from the Dogtag service on the IPA server. Like many other services, Nginx wants the certificate in privacy-enhanced mail (PEM) format in two files for the certificate and the key. If the server is named *www.mykier.ip*, you will receive a *www*.

IT Highlights at a Glance



Too busy to wade through press releases and chatty tech news sites? Let us deliver the most relevant news, technical articles, and tool tips – straight to your Inbox.

Linux Update • **ADMIN Update** • **ADMIN HPC**

Keep your finger on the pulse of the IT industry.

ADMIN and HPC: bit.ly/HPC-ADMIN-Update

Linux Update: bit.ly/Linux-Update

mykier.ip.crt file and a www.mykier.ip.key file in the response. The long detour by a local Network Security Services (NSS) database is no longer needed thanks to Certmonger. When a certificate is approaching its expiration date, you can renew it with the -r (renew) option. For the Nginx server to access the files, it of course needs the appropriate permissions:

```
chown -R nginx. /etc/nginx/cert/*
```

In the default /etc/nginx/nginx.conf configuration file, simply enable the commented out # Settings for a TLS enabled server section and add the reference to your certificate:

```
ssl_certificate 2
    "/etc/nginx/cert/www.mykier.ip.crt";
ssl_certificate_key 2
    "/etc/nginx/cert/www.mykier.ip.key";
```

After restarting the service, your web server now also responds to HTTPS requests with the IPA-signed certificate. You can then use the same principle to create other services (e.g., IMAP, SMTP, or even MQTT) and generate suitable certificates.

Building Trust Relations

For your connected clients to accept the certificates created in this way, they need to trust the CA on your IPA server, where you will find the root CA in /etc/ipa/ca.crt. The method you need to import a custom CA depends entirely on the application. For example, the Firefox browser lets you store ca.crt in a specific directory. On Windows, this would be

```
- "%USERPROFILE%\AppData\Local\2
    Mozilla\Certificates"
- "%USERPROFILE%\AppData\Roaming\2
    Mozilla\Certificates"
```

and on Linux:

```
- "/usr/lib/mozilla/certificates"
- "/usr/lib64/mozilla/certificates"
```

On Linux systems with SELinux enabled, it is again important for the certificates directory and the contained files to have the correct cert_t context; otherwise, Firefox cannot read them. Applications such as Redis either pass the path in to ca.crt at the command line or save it in the configuration file. Today, many applications use the operating system’s CA trust.

On an EL8 or Fedora Linux system, copy the ca.crt from your IPA server to the /etc/pki/ca-trust/source/anchors directory (don’t forget the SELinux context) and run the update-ca-trust command as root or with sudo. Applications such as the Chrome or Chromium browser use the system’s trust chain and do not need an individual configuration. On Windows, you can import your own certificates with a Group Policy or a PowerShell script.

Conclusions

FreeIPA with the integrated Dogtag services and Certmonger client greatly simplifies certificate management on the intranet. Gone are the days of local NSS databases and tedious workflows to copy requests and their responses back and forth – along with the need to convert the resulting PKS12, PEM, KEY, or CRT files somehow to the right formats. The open source toolset provides you a quick option for creating and deploying certificates for all TLS-enabled services in just a few simple steps. ■

- Info
- [1]

Dogtag: [\[https://www.dogtagpki.org/wiki/PKI_Main_Page/\]](https://www.dogtagpki.org/wiki/PKI_Main_Page/)
- [2]

FreeIPA: [\[https://www.freeipa.org/page/Main_Page\]](https://www.freeipa.org/page/Main_Page)

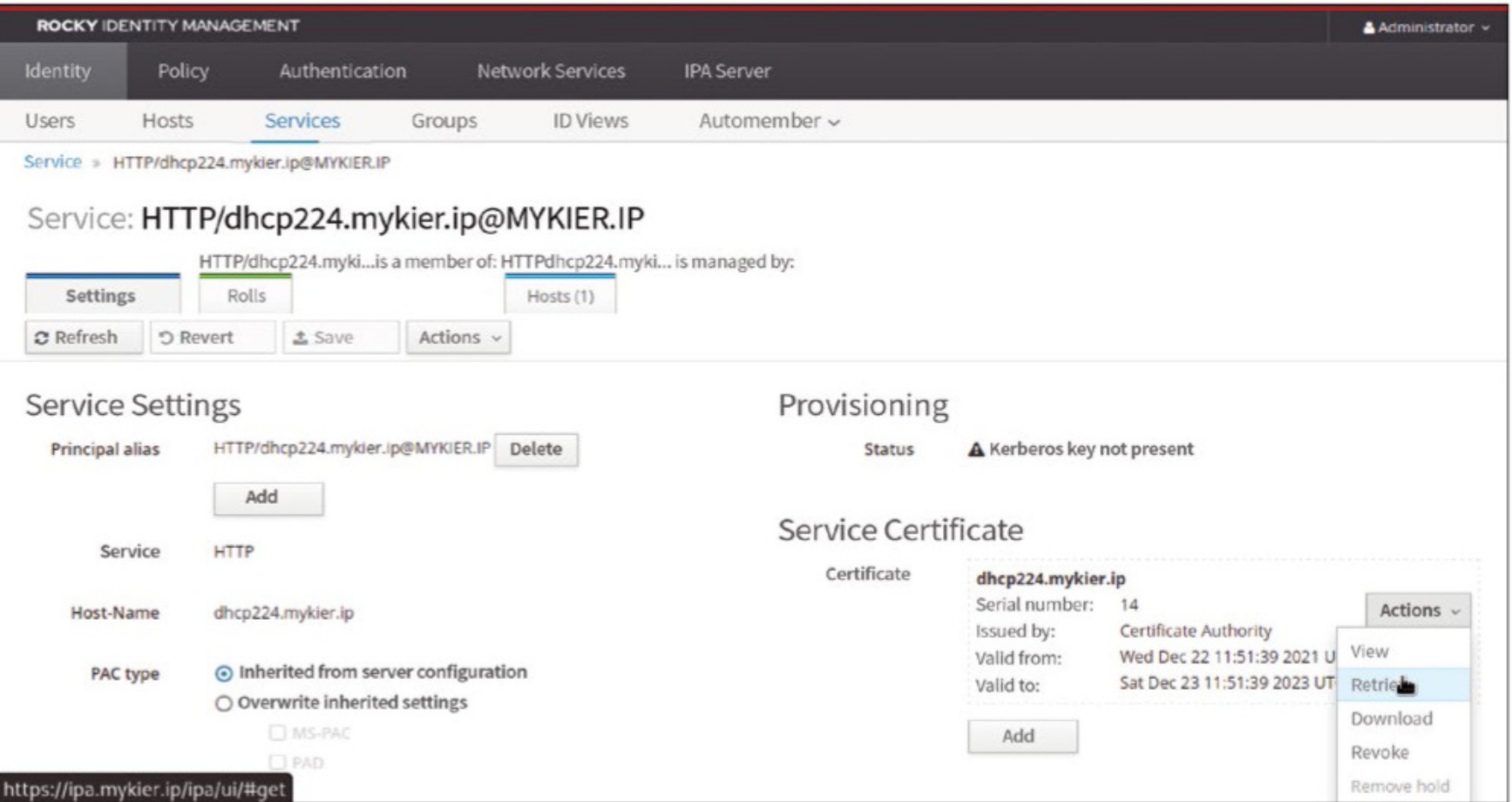


Figure 3: Issuing certificates for individual services and machines.



FOSSLIFE

Open for All

**News • Careers • Life in Tech
Skills • Resources**

FOSSlife.org



Detect anomalies in metrics data

Jerk Detector

Anomalies in an environment's metrics data are an important indicator of an attack. The Prometheus time series database automatically detects, alerts, and forecasts anomalous behavior with the Fourier and Prophet models of the Prometheus Anomaly Detector. By Martin Loschwitz

Attacks on environments are just as much a part of the daily grind in IT as operating the IT infrastructure itself. The range of attacks is wide and depends on the attacker's goals. Classic denial-of-service attacks are not complex and quite easy to detect. However, when the focus shifts to sniffing data, the methods are far more subtle, and highly complex IT attacks on different levels are no longer challenging.

As complex as the attack scenarios are, one factor remains the same: Administrators want to notice as early as possible that bad things are going on in their setups so they can react promptly. The sooner an attack is detected, the sooner it can be counteracted and the less damage it can cause.

Rigid Limits of Limited Use

The ability to detect an attack early depends on the tools available and how you use them. In the past, most admins relied on run-of-the-mill event monitoring with thresholds: If the incoming data volume exceeded a certain limit, the monitoring system sounded an alarm. If too many invalid login attempts appeared in the servers' authentication logfiles, you were notified. The focus here is on

enabling you to act as quickly as possible in a specific case (i.e., conveying the current situation).

This approach is not particularly up to date or smart. Modern monitoring systems like Prometheus collect such large volumes of metrics data that it can be used to identify trends and anomalies, potentially indicating that attacks are in progress. Even distributed denial-of-service (DDoS) attacks have ceased to follow the principle of taking a server offline with as much traffic as possible in as short a time as possible. Instead, postmortem analyses of attacks regularly reveal that attackers successively increased the traffic in the weeks leading up to an attack and did so in such a way that they always flew under the radar of the thresholds in monitoring. At the decisive moment, a relatively small peak in the attack volume was the final straw that broke the servers' backs. With better trend analysis (e.g., with the help of Prometheus), such attacks become quite predictable.

Gaussian Z-Scores

The statistical Z-score plays an important role when it comes to detecting anomalies, allowing you to define

what an anomaly is in the context of a particular environment. Large infrastructures, for example, will apply far higher thresholds for DDoS than websites with only a few visits per day. From your point of view, anomaly detection now means finding a reliable mean value for individual datapoints and then defining limits within which the current measured values are allowed to deviate from the norm. The "cry wolf" effect of permanent false positives should not be underestimated. Sooner or later, no one will take a monitoring system seriously if it constantly sounds the alarm without reason. Instead of a blunt weapon, a fine scalpel comes into play when detecting anomalies in metrics data, and the Z-score is a prime example of a particularly good scalpel.

A little excursion into the world of Carl Friedrich von Gauss's mathematics is unavoidable. Most people have probably heard of Gaussian normal distribution. Simply put, Gaussian theory states that, for any number of measured values, the extremes occur rarely and the median (i.e., the 50th percentile) occurs particularly frequently. On both sides of the x -axis, the number of matches per value increases as the median is approached. Given 100 servers, the power

consumption of most devices is likely to fall around the median, with a few individual machines requiring particularly greater or very little power. These values form the outer extremes of an imaginary chart with all measured datapoints.

Percentiles generally play an important role in calculating the Z-score. The first step is to calculate the median, which is the 50th percentile (i.e., 50 percent of all measured values correspond to this value). The Z-score is used to find out how far a single datapoint deviates from this median and is calculated by

$$\text{Z-score} = (\text{value} - \text{median}) / \text{standard deviation}$$

which can be either defined with generic values or determined individually. Common values are the 68th percentile (i.e., in a dataset of 100 values, 68 of those fall within ± 1 standard deviation [SD] of the mean), the 95th percentile (± 2 SD), and the 99.7th percentile (± 3 SD).

Red Hat with Groundwork

Now the question arises as to what you need to do to generate appropriate alerts from your metrics data, which is a practical possibility only if you use a time series database (e.g., Prometheus). Prometheus collects the metric values from “exporters” on the target systems and stores them centrally. This data can be evaluated by a custom query language, and Grafana can display the Prometheus data graphically. Prometheus generates alerts with its alert manager component if individual metrics assume certain values or are outside of defined limits.

The question of how to use an existing Prometheus installation for effective anomaly detection is provided by Red Hat. The Prometheus Anomaly Detector (PAD) comprises several components designed to detect anomalies from historic data on the one hand and machine learning and projection on the other.

A look under the hood shows the combination of components. Red Hat

generally assumes that you will roll out PAD as a component in OpenShift, although this scenario is not enforced. However, if you want to use PAD, you will need an environment comprising Prometheus, Alertmanager, and Thanos – more about this in a moment – because PAD does not seek to be a monitoring tool itself, but to dock onto existing setups. At its core, PAD is a Python application that applies two Python libraries for artificial intelligence (AI) and machine learning (ML): Fourier and Prophet.

Fourier and Prophet

Metrics are available as numbers in Prometheus; however, sinusoidal curves are far better suited for accurate analysis of the current data, as well as the future development of these values in the context of machine learning. Each value is represented in the form of a frequency. The Fourier algorithm is responsible for the transformation between the worlds by using sine and cosine to convert the numbers into frequency signals, after which useful basic data is available for the prediction. This process is known as “fast Fourier transformation,” which goes back to the mathematicians James Cooley and John Tukey, who popularized the idea for converting data into sinusoidal curves in a paper that appeared in 1965 [1]. The method is

now considered a standard algorithm of modern IT. The Fourier module in Python has an AI-trainable model to predict the further evolution of an existing graph on the basis of historical values.

The other algorithm that PAD uses to detect anomalies, Prophet, comes directly from the Facebook social network (Figure 1) and is significantly more complex in direct comparison with Fourier. In return, it allows factoring into its predictions such things as the potential seasonality of data, including the factors year, week, and day. All told, the dataset that Prophet uses to analyze ongoing data streams, predict their continuation, and raise the alarm if necessary is far larger. PAD, with its two implementations of anomaly detection, goes significantly further than the analysis of the acquired metrics described at the beginning. The aim is not just to notice that something is wrong at an early stage, but to notice it even earlier: Both Fourier and Prophet are trainable AI models in PAD that use existing metrics data to predict the evolution of the respective metrics, allowing for a response at the slightest sign of an anomaly. Your task is to compare the developments of a metric value calculated by Prophet or Fourier with the actual state. Ideally, thanks to the trained models in PAD, you will notice very quickly that something suspicious is going on (Figure 2).

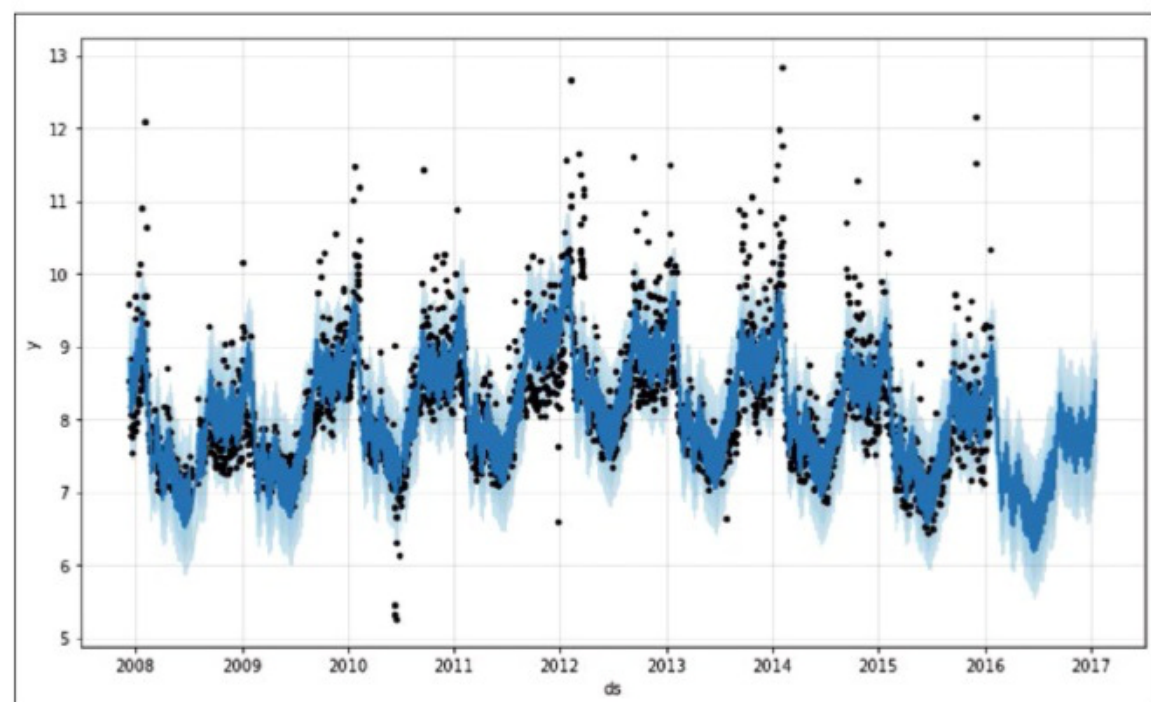


Figure 1: The Prophet PAD component is trimmed to respond correctly with predictive methods to the smallest deviations in metric data, represented here by the black dots.

A Failure to Scale

From the very beginning, the Prometheus developers designed their work to ignore high availability in the classical sense. Instead, you have to run multiple instances of Prometheus at the same time. Because retrieving metrics data from the target systems ties up virtually no resources, it doesn't generate more load on the monitored systems. If one of the running instances fails, the idea is that you have enough other instances to query. In this way, comprehensive deduplication is implemented at the alert manager level: If 20 instances feed an alert into the alert manager because of the same metric value, the alert manager still sends only one message to the alerting targets.

However, this use case turns ugly when it comes to horizontal scaling. Massively scalable environments, in particular those with hundreds or thousands of hosts, mean that individual Prometheus instances become a bottleneck over time. If you go for manual sharding, however, you lose the central advantage of the single point of administration because individual Prometheus instances then query a local list of targets, and it's your task to connect to the appropriate Prometheus instance to view the

metrics for a particular host. What's more: Grafana also needs to be configured in the same way and different queries need to access the different data sources in Grafana. What's almost worse, though: The more data Prometheus stores, the slower and more unresponsive it becomes. However, you do have a legitimate interest in retaining historical data because it allows for better scalability planning and makes it easier to identify trends. If you keep too much legacy information in Prometheus, calling individual Grafana dashboards will soon take several seconds. Intelligent downsampling of data is missing, as is high-performance storage for archived data outside of Prometheus itself.

Thanos and Historical Data

Thanos is now a separate project, independent of Prometheus, that wraps around several self-sufficient Prometheus instances. First, Thanos provides a unified query view: No matter which of the Prometheus instances connected to Thanos has the metrics data for a particular host, the admin always talks to Thanos, which gathers the data appropriately in the background. The same thing also applies to Grafana instances.

Second, a Thanos component is responsible for storing historical data in a meaningful way. To this end, Thanos provides an interface, known as StoreAPI, to its own storage implementation, as well as interfaces to other databases. Thanos provides Prometheus with the Sidecar component, which dynamically handles writes not to local Prometheus memory, but distributed across the network. The long-term storage of the historical data is so much better than if Prometheus itself were used for this purpose.

In brief, Thanos is a practical extension to the plain vanilla Prometheus that has found its way into many installations around the world. The fact that Prometheus anomaly detection takes advantage of Thanos is hardly surprising: In particular, storing historical data is extremely helpful when it comes to detecting anomalies.

PAD in Practical Use

As mentioned earlier, PAD is designed to run in OpenShift, but the container runs quite well without Red Hat's orchestrator. The following example assumes that you use Podman to manage your containers. To begin, source the complete PAD container; the code is also available on GitHub [2]:

```
podman pull quay.io/aicoe/prometheus-anomaly-detector:latest
```

PAD gets the metrics for which it is supposed to detect anomalies per Fourier and Prophet via environment variables.

Next, you need to build the command line to start the container. PAD gets the metrics for which it is supposed to detect anomalies with Fourier and Prophet from environment variables. These, in turn, can be easily passed in to the container. The command

```
docker run \
  --name pad -p 8080:8080 --network host \
  --env FLT_PROM_URL=http://pad.local.lan:9090 \
  --env FLT_RETRAINING_INTERVAL_MINUTES=15 \
  --env FLT_METRICS_LIST='up'
```

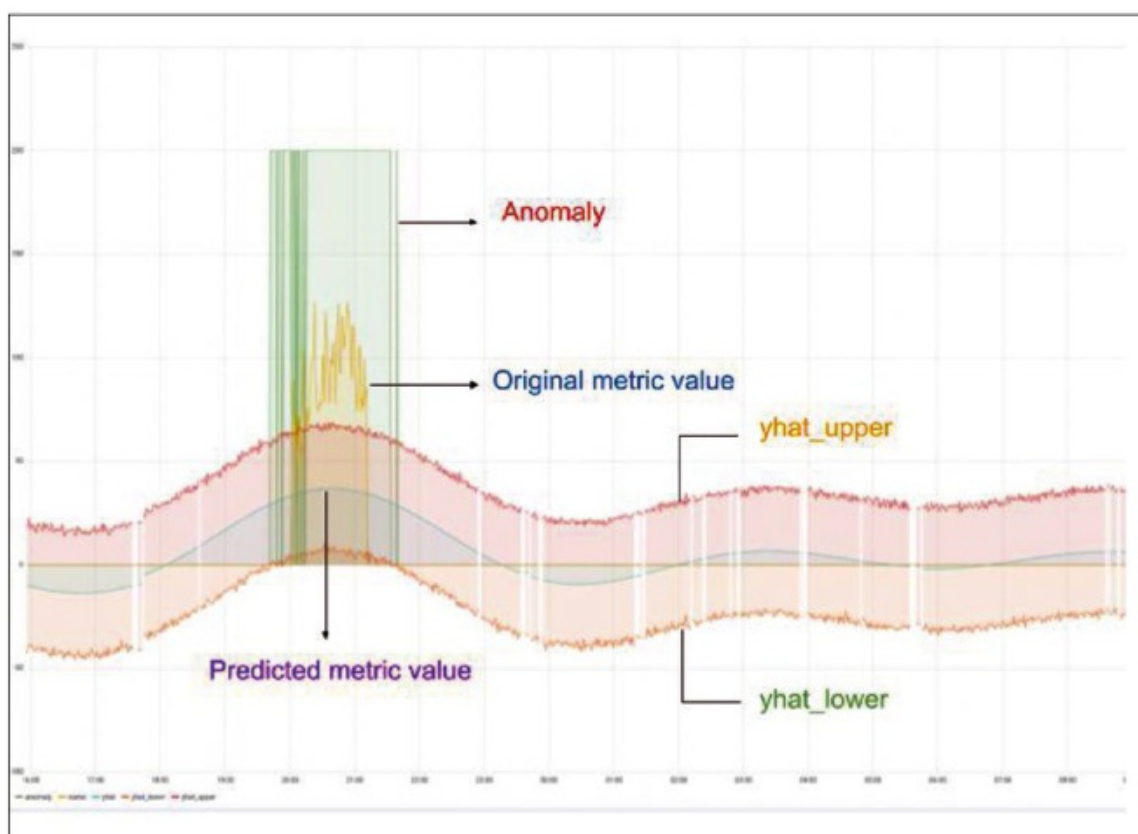


Figure 2: The blue line shows the predicted evolution of the data for the respective metric; yellow and green indicate the range of variation. The anomaly is clearly visible.


```
--env APP_FILE=app.py 2
--env FLT_DATA_START_TIME=3d 2
--env FLT_ROLLING_TRAINING_WINDOW_SIZE=2
15d quay.io/aicoe/2
prometheus-anomaly-detector:latest
```

starts the container and sets up as a metric in `FLT_METRICS_LIST` (i.e., you want to know whether or not the systems are running). Instead of up, you need to add the names of the metrics for which you want to detect anomalies. If you enter the Prometheus Node Exporter's `node_filesystem_avail_bytes` metric here, for example, you are telling PAD to monitor changes in the allocation of storage drives of individual devices, because suddenly increasing space occupation (i.e., reductions in free space) can be an indicator of some undesirable processes that justify a closer look. PAD alone usually does not help you, because Prometheus does not visualize the data graphically. Grafana is the tool you want to use, and the PAD developers make it easy to do just that, because PAD exports the calculated metrics data in Prometheus format. An existing Prometheus instance can read the data from PAD just as from any other

exporter. For this step, the developers rely on Flask.

The rest is plain sailing. Once the metrics data and the predictions are available in Prometheus, dashboards can be created in Grafana in the usual way. What's more, the predictions from Fourier and Prophet can be integrated into the same dashboards and superimposed – together with the measured values, if required (Figure 3). If you want to set up alarms from the predictions, you can do so in the alert manager. A Red Hat talk from 2019 [3] provides some details of the configuration.

Conclusions

Administrators often turn up their noses when vendors present their AI solutions for attack detection; in fact, it's not uncommon to hear them referred to as hocus pocus. However, Red Hat has come up with a very concrete and immediately usable approach to generating added value in everyday life with AI in the form of PAD. The more metric data Fourier and Prophet have available, the better they can train their models and the more reliable the predictions become. Therefore, you do not need

to allow for start-up time with false positives at the beginning. However, the extra work will pay off when you track down an attacker because you noticed even earlier than usual that something was wrong. ■

Info

- [1] Cooley, J. W. and J. W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation*, 1965;19:297-301, [<https://www.ams.org/journals/mcom/1965-19-090/S0025-5718-1965-0178586-1/S0025-5718-1965-0178586-1.pdf>]
- [2] Prometheus Anomaly Detector: [<https://github.com/AICoE/prometheus-anomaly-detector>]
- [3] AIOps: Anomaly detection with Prometheus, by Marcel Hild, Linux Foundation, [<https://events19.linuxfoundation.org/wp-content/uploads/2017/12/AIOps-Anomaly-Detection-with-Prometheus-Marcel-Hild-Red-Hat.pdf>]

The Author

Freelance journalist Martin Gerhard Loschwitz focuses primarily on topics such as OpenStack, Kubernetes, and Ceph.

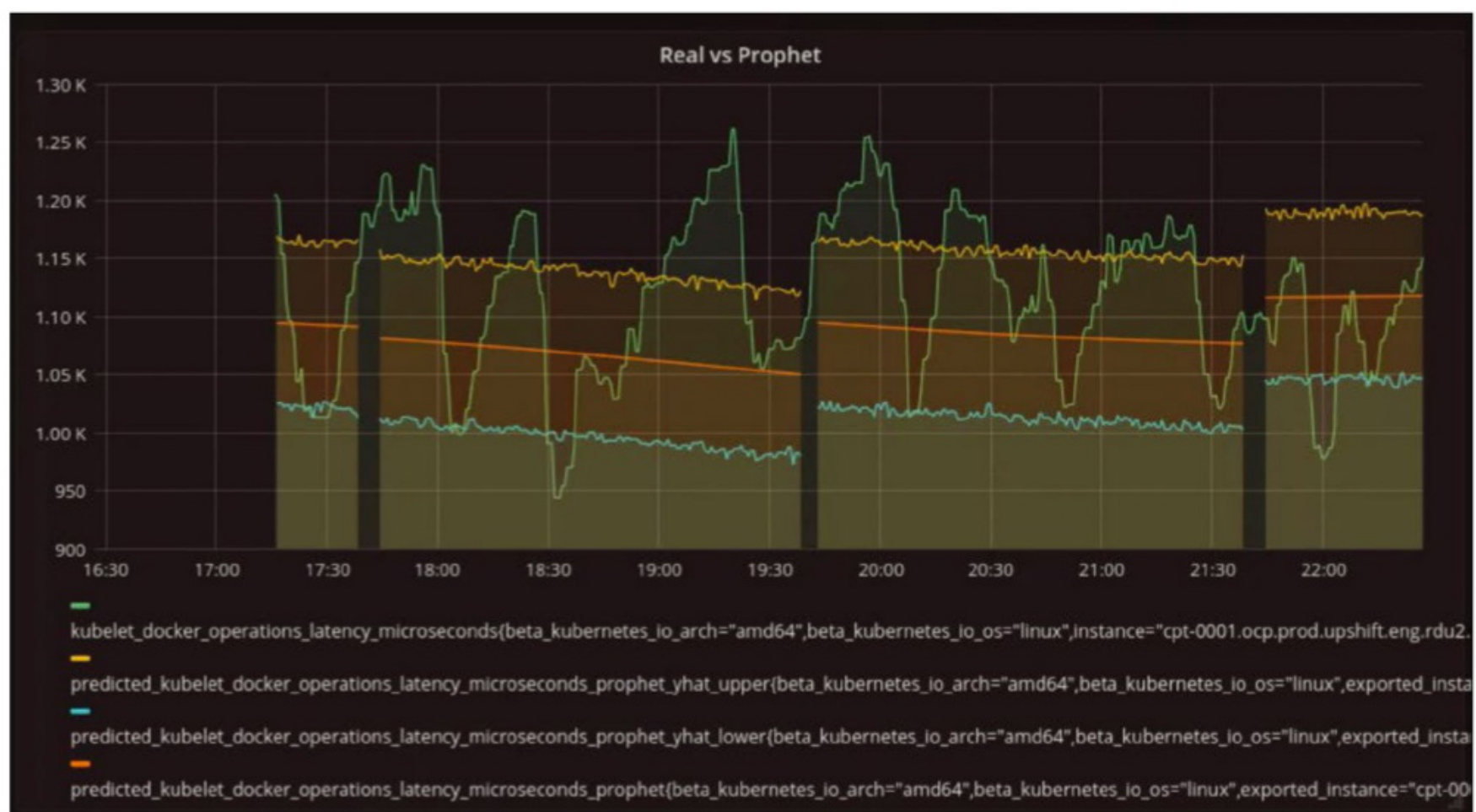


Figure 3: Once the data from PAD has found its way into Prometheus, it can be visualized in Grafana, which is where you see a comparison between the data and the Prophet predictions.



Puppet Bolt orchestration tool

Lightning Strike

Puppet Bolt free software automates administrative tasks to speed up the admin's daily work. By Holger Reibold

Because Bolt is a member of the Puppet product family, the question naturally arises as to how Bolt [1] differs from Puppet. Puppet is used for continuous resource management. In particular, it provides monitoring functionality and checks at short intervals whether the services in question are still available or whether infrastructure elements have gone missing. Bolt basically zooms in on point-in-time changes. Instead of using declarative statements that define an infrastructure, Bolt is more about when commands are executed and which ones. In particular, the tool simplifies the execution or orchestration of tasks.

Admins benefit from the ability to run a script over any number of network nodes. Bolt uses plans that bundle the execution details. The focus is particularly on error handling, but comparatively simple

scripts can also be used to handle complex tasks.

Overview

Ad hoc commands and scripts are run on the infrastructure with the Puppet Enterprise (PE) orchestrator or with Puppet's standalone task runner, Bolt. Bolt lets you patch and update systems and services, troubleshoot servers, roll out applications, and start and stop services. It runs on a standard workstation (Linux, Windows, macOS), and secure shell (SSH), secure copy (SCP), Windows Remote Management (WinRM), and other popular authentication methods (password, public key) connect to the remote node. According to the developers, the solution scales to more than 1,000 simultaneous connections. Bolt uses YAML files or its own

orchestration script wrapper, called a "plan." Above all, if statements are used in the scripts for concrete error handling. Administrators who are already familiar with YAML files can generate their tasks in this particular format and then use Bolt's built-in tool to convert YAML files into Bolt plans. You will find various special features beneficial: Bolt has pre-built scripts that you only need to adapt to specific tasks; it also lets you use existing automation scripts and offers support for Python, Ruby, and PowerShell. Although many orchestration tools rely on agents, Bolt also supports agentless deployment or a combination of the two strategies. Bolt also offers Bash support and workflow orchestration.

Installation and Setup

Bolt runs on all popular operating systems. In addition to a Linux-based

Photo by lee junda on Unsplash

machine, you can use the orchestration tool on a macOS or Windows workstation. Installing Bolt on Debian is a matter of a few simple commands:

```
wget https://apt.puppet.com/2
  puppet-tools-release-bullseye.deb
sudo dpkg 2
  -i puppet-tools-release-bullseye.deb
sudo apt-get update
sudo apt-get install puppet-bolt
```

To run Bolt on a macOS system, you first need to install Homebrew [2], an open source package manager for the operating system. To install Homebrew, run the following command in the macOS terminal:

```
/bin/bash -c "$(2
  curl -fsSL 2
  https://raw.githubusercontent.com/2
  Homebrew/install/HEAD/install.sh)"
```

Use the `tap brew` command to instruct Homebrew to use additional repositories. By default, the `tap` command assumes you are accessing sources from GitHub repositories, so you need to prepare Homebrew for using the Puppet sources by typing:

```
brew tap puppetlabs/puppet
```

To install Bolt, run the command:

```
brew install --cask puppet-bolt
```

Alternatively, you can use the macOS installer and use the DMG file from the Bolt project site.

To use Bolt on Windows, you need Chocolatey [3], a package manager that performs typical functions such as downloading and installing applications. To install the Bolt packages, and refresh the environment, run the commands:

```
choco install puppet-bolt
refreshenv
```

To import the Bolt PowerShell modules, type

```
Install-Module PuppetBolt
```

and run a Bolt cmdlet as a test. Ideally, you will not see any error messages. If you do, you might need to add more Bolt modules to PowerShell or edit the execution authorizations.

Task-Specific Configuration

Bolt offers a wide range of customization options for global and project-specific configuration. Four categories can be distinguished:

- Customizing Bolt's general behavior, such as choosing the format for displaying the output and defining the number of threads for connecting to targets
- Defining project-specific settings by specifying how to deal with concrete orchestration tasks, including configuring the path to an inventory file or to a Hiera configuration file. (Hiera is a key/value database for the configuration data.)
- Deciding which transport protocols to use, such as adjusting the path to your private SSH key or the port for the WinRM connection
- Grouping inventory data by targets and assigning them their own configurations

Bolt options and functions are configured at the project, user, or system level. At the project level, you specify the Bolt configuration in the `bolt-project.yaml` and `inventory.yaml` files. Customizations at the user and system level are defined in `bolt-defaults.yaml`. If the specific use case does not require user-specific or global configurations, configuration at the project level is the recommended approach.

Creating a Bolt Directory

Your tasks in Bolt start with creating a Bolt project and setting up the targets. A Bolt project is a directory that includes the project-specific configuration settings. The first step is to create the project directory and convert it into a Bolt project:

```
mkdir first_bolt_project
bolt project init first_bolt_project
```

Bolt requires a specific directory structure for the projects – plans and tasks will not work without this structure. In this context, the directory structure of a Bolt project is closely linked to Puppet modules. For example, if you plan to install Apache on a remote system, you need to create an Apache module directory. To stay with this example, you need to generate a module directory along with an Apache subdirectory by changing to the project directory and creating two directories, one for the project files and one for the plans:

```
cd first_bolt_project
mkdir -p module/apache/plans
mkdir -p module/apache/files
```

The directory structure for the `first_bolt_project` folder looks like this:

```
|-- bolt-project.yaml
|__ module
    |-- apache
        |-- files
        |-- plans
```

Defining Targets

The next step is to define the targets. In this section, I assume that you want to run an Apache web server on a Docker installation and perform typical tasks there. The connection between the Bolt installation and the remote systems is set up with either SSH or WinRM. As a rule, SSH will be the better choice. To talk to a Docker installation, generate a file named `Dockerfile` in the root directory of the Bolt projects and assign it the following commands:

```
from rastasheep/ubuntu-sshd
run apt-get update && 2
  apt-get -y install libssl-dev
expose 80
cmd ["/usr/sbin/sshd", "-d"]
```

The Dockerfile defines an Ubuntu container, including the SSH service, which lets you talk to the entities involved. The next step is to create the `docker-compose.yaml` file, which generates two container instances. To do

this, create the file in your project directory and assign the settings shown in [Listing 1](#).

Now with the Compose file and the help of the Dockerfile, create two containers named `target1` and `target2`. The SSH connections point to ports 2000 and 2001, respectively, and the HTTP connections to ports 3000 and 3001. To create and run the Docker containers and make sure the two containers are running, use the commands:

```
docker-compose up -d --build
docker-compose ps
```

Screen output should appear telling you that the Docker containers are ready to use, which means you can now run commands against the containers.

Commands Against Targets

Before you start executing extensive plans, you will want to run various commands against the target – not least to familiarize yourself with Bolt specifics. The general syntax for command execution is:

```
bolt command run <command>
--targets <target name> <options>
```

Listing 1: docker-compose.yaml

```
version: '3'
services:
  target1:
    build: .
    ports:
      - '3000:80'
      - '2000:22'
    container_name: target1
  target2:
    build: .
    ports:
      - '3001:80'
      - '2001:22'
    container_name: target2
```

Listing 2: Output from a Running Plan

```
Starting: plan apache::install
Starting: task package on target1, target2
Finished: task package with 0 failures in 18.00 sec
Finished: plan apache::install in 20.00 sec
Plan completed successfully with no result
```

To execute the `whoami` command on target 1, for example, you would type:

```
bolt command run whoami -t 127.0.0.1:2000
-u root -p root --no-host-key-check
```

This command targets a system with IP address 127.0.0.1 and port 2000, through which SSH is addressable. The command also passes in a username and password to enable access to the system. The `--no-host-key-check` option disables certificate authentication. Typical output for the command would be:

```
Started on 127.0.0.1:2000...
Finished on 127.0.0.1:2000:
  STDOUT:
    root
Successful on 1 target: 127.0.0.1:2000
Ran on 1 target in 0.3 sec
```

The output indicates that you have successfully run your first Bolt command on a target system. However, you will typically want to address a whole group and not just one system. To do this, Bolt uses an inventory file that lets you group an arbitrary number of systems.

Deploying Bolt Plans

Plans let you link commands, scripts, and tasks, combining them to create powerful workflows. Basically, you can use Puppet’s own language or YAML for your plans. The next example installs an Apache web server on the Docker targets created previously. It also takes care of starting the Apache services and uploads a simple home page. To begin, you need some installation instructions for the Apache web server. In the `apache` subdirectory, create the `plans` directory and the `install.yaml` installation script:

```
parameters:
  targets:
    type: TargetSpec
steps:
  - name: install_apache
    task: package
    targets: $targets
```

```
parameters:
  action: install
  name: apache2
  description: "Installs Apache"
```

In the first section of the Bolt plan, you need to define the parameters that your plan will accept. In this example, it is the `TargetSpec` type, which you use to pass multiple targets to a plan. For the parameters specified in the plan to be used, you need to pass in the `--targets` option at the command line during execution. The `steps` section is where you specify the main part of your plan. In this example, it is named `install_apache` and uses the Bolt package task to install Apache on the target systems. Bolt makes generous use of these predefined task configurations – in particular, reused tasks are defined there. The plan also defines the actions to be performed (installation) plus the name of the package. Having the `install.yaml` file in place in the `plans` subdirectory is important. To run the plan in a container group, use:

```
bolt plan run apache::install -t containers
```

The plan is designated by two segments separated by a double colon. The first segment specifies the name of the module in which the plan resides, and the second segment specifies the plan file, but without the file extension. In this example, the plan is located in the `Apache` modules directory and is named `install.yaml`. The name of the plan is therefore `apache::install`. You can watch your plan running at the console. Typical output is shown in [Listing 2](#). In a practical use case, it makes sense to use Bolt to install Apache on your target systems. The orchestration software offers a solution. To begin, create a script file in which you store the specific startup parameters for the Apache startup process. Drop this script (i.e., `start_apache.sh` in this example) in the `files` subdirectory below the Apache module directory. Next, add the following block to your `install.yaml` file:


```
- name: start_apache
  script: apache/start_apache.sh
  targets: $targets
  description: "Start Apache Services"
```

The next time the plan is executed, the Apache install will also start, basically clearing the way to access the various Apache web servers. Note that Bolt assigns port 3000 to the first target, port 3001 to the second, and so on. In this example, the default home page is 127.0.0.1:3000 for the first target.

To store your HTML pages on the various Apache installations, create a simple HTML file, name it `index.html`, and store it in the `/apache/files` directory of your plan configuration. Before you can upload, you first need to create a `src` parameter of the String type in the parameter configuration. The extended configuration then looks like this:

```
parameters:
  targets:
    type: TargetSpec
  src:
    type: String
```

Now extend the `steps` section, adding the following block of code:

```
- name: upload_homepage
  upload: $src
  destination: /var/www/html/index.html
  targets: $targets
  description: "Upload the Homepage"
```

After running the plan again, you should see the new homepage when you access `127.0.0.1:3000` (i.e., the address of the Apache installation).

Advanced Bolt

One key feature of Bolt is its modules, which bundle plans and tasks into typical workflows to facilitate integration. Another benefit is that you can share the modules with third parties; therefore, you can perform identical actions on external networks. However, if you plan to use elements, note that the functionality of the modules often depends on other modules. If you install the module from the Bolt console, Bolt automatically manages these dependencies for you. To do so, the module is added to the `module` section of the project configuration file (`bolt-project.yaml`).

In the next step, Bolt resolves the modules and their dependencies and generates a Puppet file. Do not modify this file. Finally, Bolt installs the modules and dependencies in the module directory (`module-dir`), which you will find in the Bolt project directory (`.modules`).

Plugins also simplify orchestration. They support dynamic loading and modification of information at Bolt runtime, which means that Bolt actions can be controlled in a targeted way. You can use three different types of plugins:

- Reference plugins are used to retrieve data from an external source and store the data in a static data object.
- Secret plugins are extensions that provide encryption and decryption facilities.
- Puppet library plugins are used when installing Puppet libraries

Commercial GUI

Bolt is managed entirely at the command line, which is unlikely to faze most administrators. If it does, the commercial Puppet Enterprise offers a convenient web interface. This license is also worth considering for large organizations that require integrated governance, more flexibility, and team-oriented workflows. It also includes the ability to scale automation features and monitoring – with and without agents.

if the associated plan uses the `apply_prep` function.

You can control the execution of the plugins with entries in the Bolt configuration files.

Conclusions

Puppet Bolt is an excellent administrative tool for orchestrating typical management tasks. Its strengths lie in its simplicity, flexibility, and ability to do without agents. For administrators who can do without the convenience of a web interface in favor of a powerful environment, Bolt is an exciting tool. However, if the lack of a web interface is an issue for you, check out the “Commercial GUI” box for fundamental details of the commercially licensed Bolt variant.

Info

- [1] Puppet Bolt: [<https://puppet.com/docs/bolt/latest/bolt.html>]
- [2] Homebrew: [<https://brew.sh>]
- [3] Chocolatey: [<https://docs.chocolatey.org/en-us/>]



Manage updates and configuration with Azure Automation

Pass Go

Microsoft Azure Automation provides a cloud-based service for handling automation tasks, managing updates for operating systems, and configuring Azure and non-Azure environments. We focus on VM update management and restarting VMs. By Thomas Drilling

Azure Automation is not just about automation tasks in and for Azure. It is a cloud-based service that provides automation features for a wide range of scenarios that can be roughly divided into three basic areas, all three of which share a number of Azure Automation features, such as schedules, modules, credentials, and certificates.

The first area covers repeatable and consistent infrastructure provisioning according to the infrastructure-as-code principle. Azure Resource Manager Templates (ARMs), Azure Bicep, and Terraform are three popular technologies that can be used for this purpose. Another large sector revolves around event-based automation, such as for diagnostics and problem resolution.

The second area is automated threat analysis, which can be performed in the context of incident detection in a security information and event management (SIEM) system – one example being Microsoft Sentinel, which comes with well over 100 third-party system collectors in

addition to many Azure- and Microsoft-specific collectors.

The third major area relates to orchestrating and integrating automation with other Azure services and third-party products. On the integration front, unsurprisingly, many Azure services already interact with Azure Automation. Even if you haven't actively dealt with the platform yet, you've probably come into contact with the service indirectly once or twice – for example, when creating a virtual machine in Azure. The *Auto-shutdown* feature in the Operations section of any Azure virtual machine (VM) is also based on Azure Automation. In this article, I look at further sample applications.

Automatic Guest System Patches

In the management section of the provisioning wizard for a new VM in the Azure portal, users have – for some time now – been able to select a number of options for patch orchestration in the Guest OS updates section (e.g.,

Automatic by OS (Windows Automatic Updates)). However, this only works for selected operating systems (i.e., Windows Server 2008 R2 SP1, 2012 R2 Datacenter, 2016 Datacenter, and 2019 Datacenter).

Essentially, Azure supports automatic guest system patching, on-demand patch assessment, and on-demand patch installation only for VMs that you create from images that have the right publisher, offering, and stock keeping unit (SKU) combination within the list of theoretically supported operating system images. Unfortunately, this means custom images or other publisher, offering, and SKU combinations are not supported.

Additionally, the VM itself must meet a number of requirements for guest system patches to work. For example, the VM in question must have the Azure VM Agent for Windows or the Azure Linux Agent installed. Furthermore, Windows VMs must run the Windows Update service for Windows virtual machines. Of course, the VM needs to be able to access the

configured update endpoints (e.g., if the VM is configured to use private repositories for Linux or Windows Server Update Services (WSUS) for Windows VMs. Of course, when creating a VM or using the REST API, you can enable guest system patching in PowerShell,

```
Set-AzVMOperatingSystem 2
```

```
-VM $VirtualMachine -Windows 2
```

```
-ComputerName $ComputerName 2
```

```
-Credential $Credential 2
```

```
-ProvisionVMAgent -EnableAutoUpdate 2
```

```
-PatchMode "AutomaticByPlatform"
```

or from the Azure command-line interface (CLI):

```
az vm create 2
```

```
--resource-group myResourceGroup 2
```

```
--name myVM --image Win2019Datacenter 2
```

```
--enable-agent --enable-auto-update 2
```

```
--patch-mode AutomaticByPlatform
```

Once guest system patching is enabled, background automation (more on this later) ensures that any critical and security patches available are

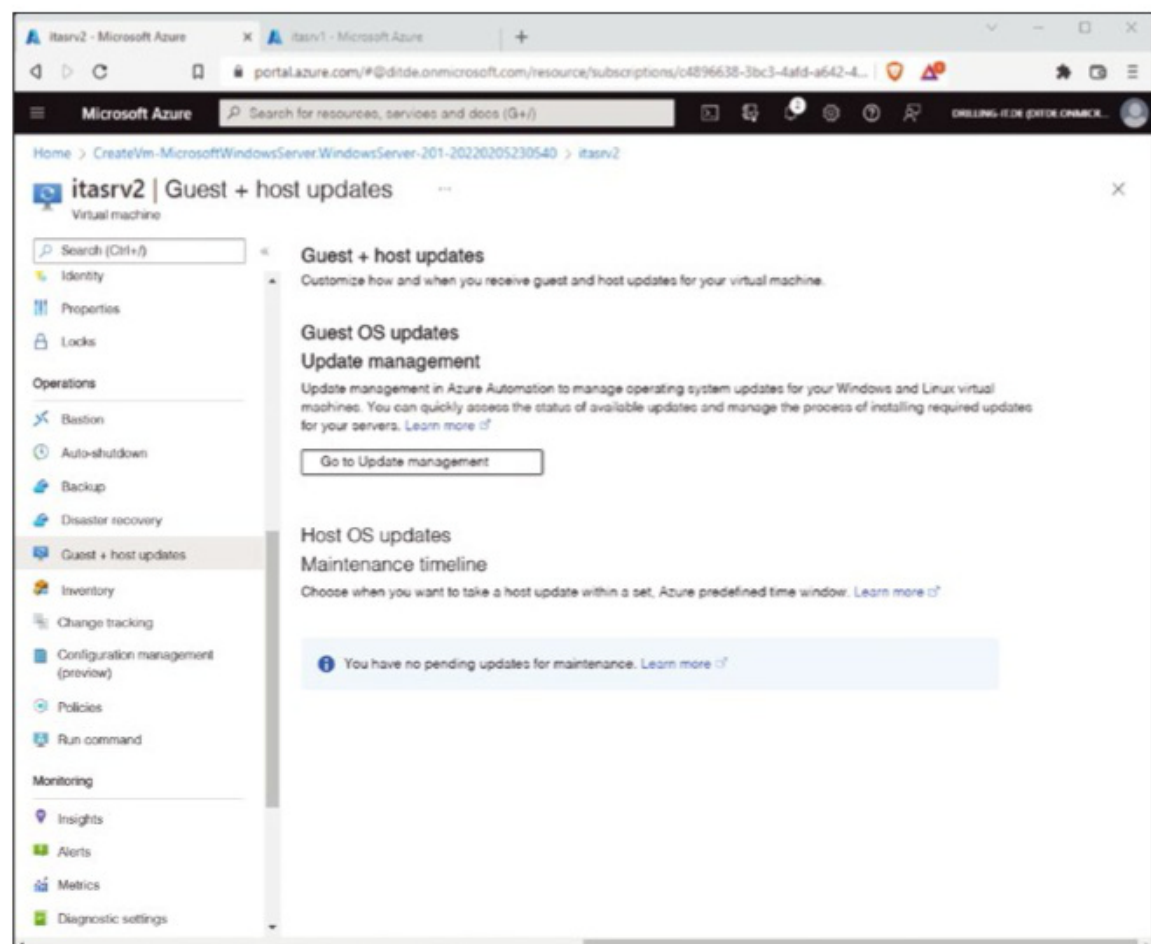


Figure 1: Update Management occupies a major part of the Azure Automation feature set.

downloaded and automatically applied to the VM. The process starts automatically every month or when Microsoft releases new patches, with patch assessment and installation taking

place automatically, although the VM might need to be restarted. To determine whether applicable patches are in place, the mechanism periodically checks each VM at 30-day intervals.

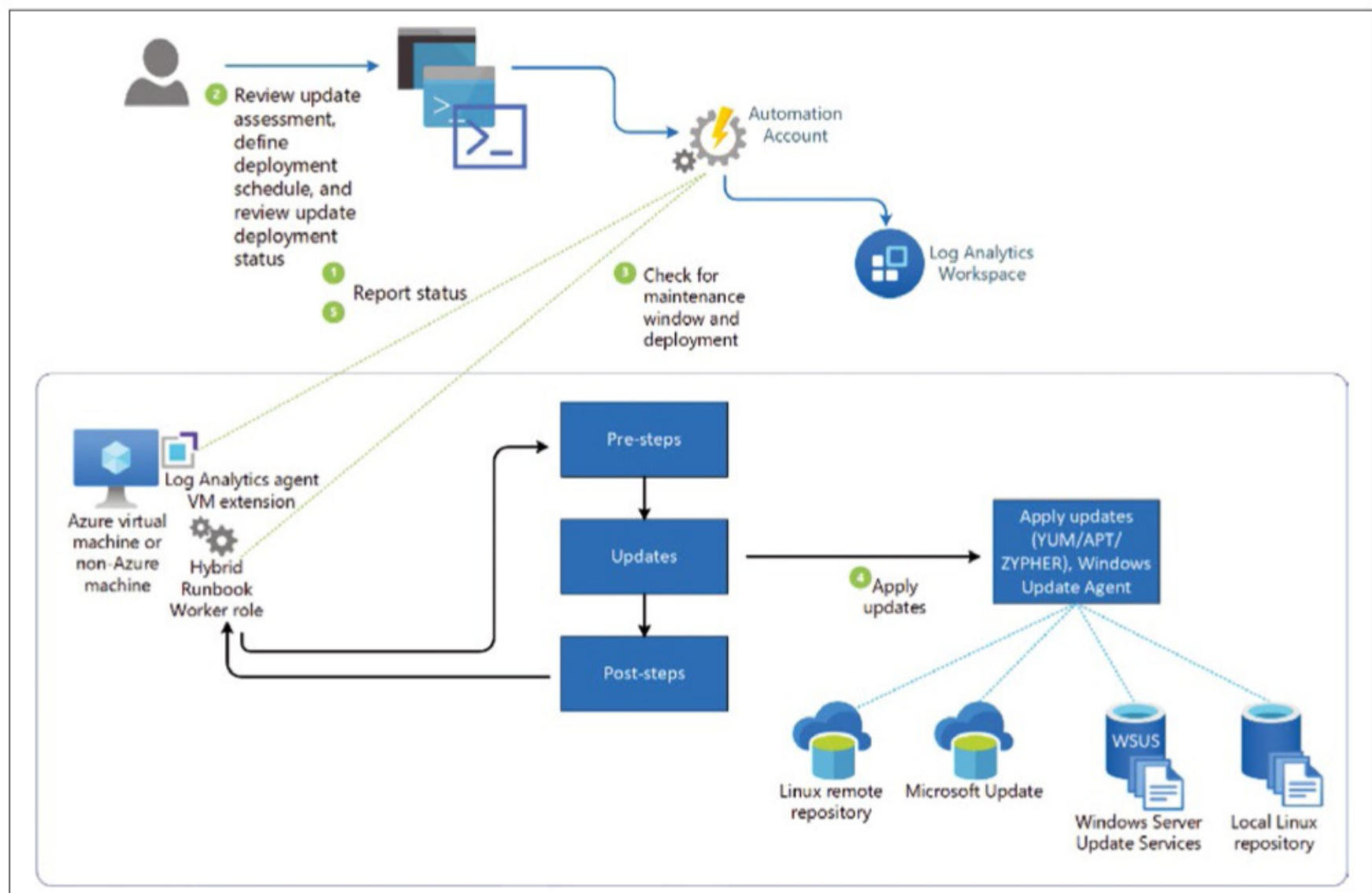


Figure 2: Interaction between Azure Automation, Windows Update, and WSUS (from Microsoft Documentation [1]).

The installation itself can happen on any day, but not at peak VM times; automatic evaluation takes care of finding the best time. For supported

guest operating systems (e.g., Windows Server 2022), you can also check the *Enable hotpatch* box at this point. The guest system is then

updated without restarting the VM. Note, however, that definition updates or other patches not classified as critical or security-relevant do not automatically reach the machine through the guest patching functionality just described. To do this – or if you want to install patches with other patch levels or in a custom maintenance window – you need the Update Management feature, which is part of the Azure Automation service. To do this, click *Guest + host updates* in the Operations section of an existing Azure VM, and then click *Go to Update management* (Figure 1).

Update Management without Azure Automation

Update Management is a feature of Azure Automation and, in principle, the service is included in the feature scope of every Azure VM. To use the service, you need an Azure Log Analytics workspace and an Azure Automation account. The cost depends on the volume of log data you store in Log Analytics. The service itself costs nothing. If you do not yet have an Automation account and a Log Analytics workspace, you can create both as part of the Update Management deployment. The VM must be switched on to do so.

Figure 2 shows how Azure's Update Management works. Windows servers obtain updates from Azure or locally from Automation Update Management. You can see that the server to

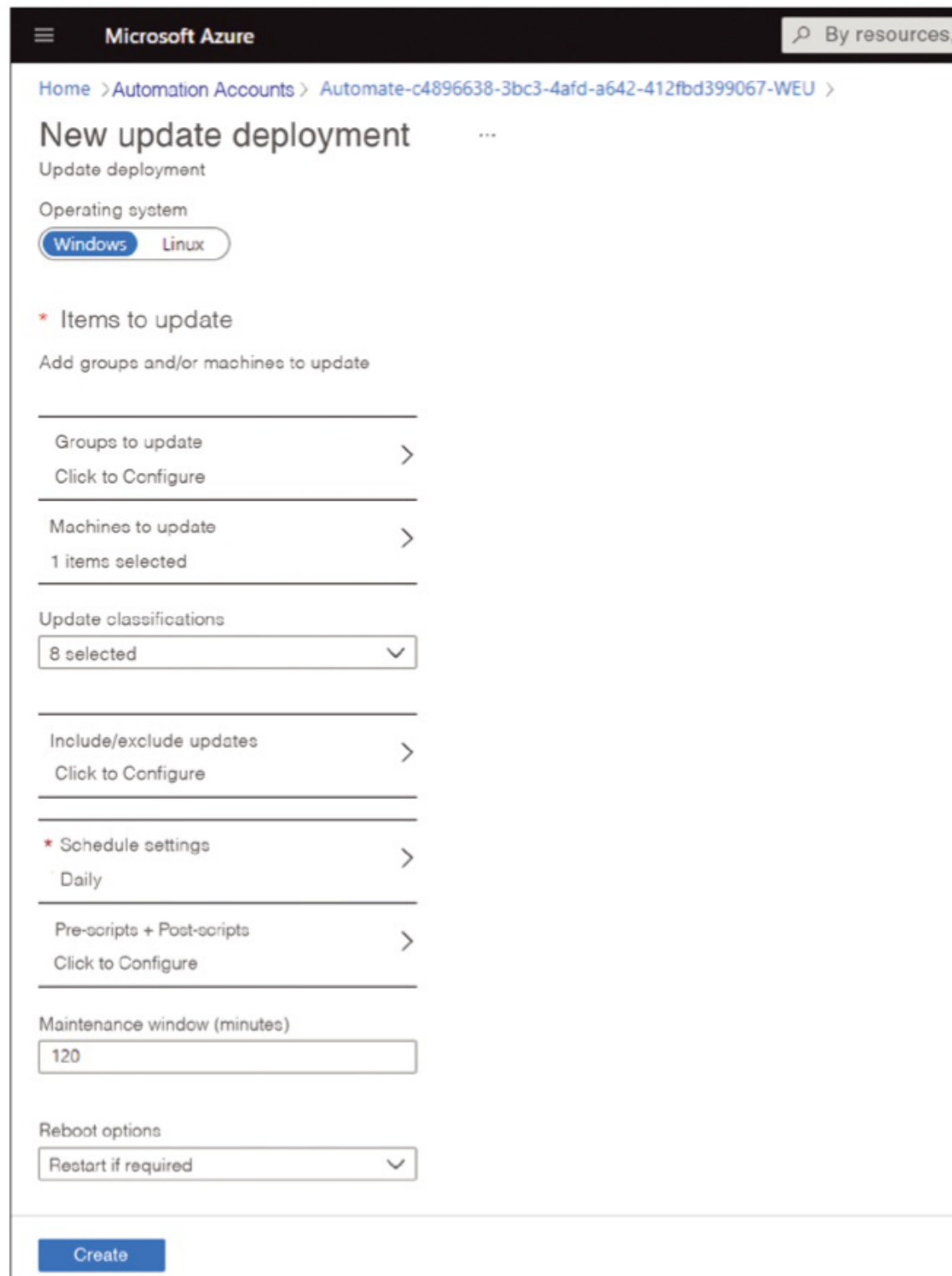


Figure 3: Setting up a new update deployment which, configured in this way, ...

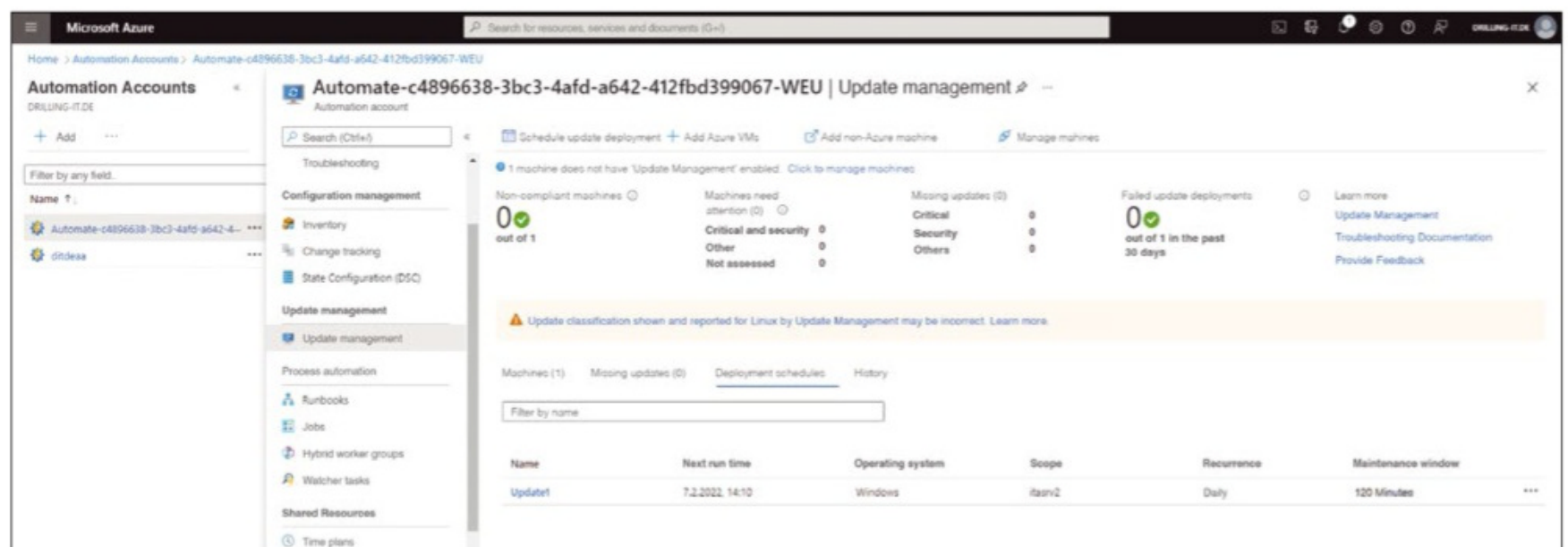


Figure 4: ... subsequently shows up under *Deployment schedules* in the Update Management section of an Automation account.

be updated – this can be an Azure VM or non-Azure VM – is connected to a Log Analytics workspace as a data source. The upper part, which relates to Azure Automation, and the lower part, which illustrates patching with WSUS or Microsoft Updates Service, are separate.

Thanks to Log Analytics integration, each server can basically write the required information to the workspace to tell you which updates it already has or which ones are missing. It gains knowledge of these states from its local Windows Update client, communicating either with Microsoft Update or with a WSUS server. For this information, the server does not absolutely need Azure Update Management.

Update management in Azure is therefore primarily used to determine the best possible time for patching and then tell the server to apply the patch. That is, Azure Update Management is used to orchestrate efficient patching. The updates are deployed by Microsoft Update or WSUS. A word of caution: If the Windows Update Agent (WUA) is configured to send messages to WSUS, the results may differ from the Microsoft Update results displayed, depending on when WSUS last synchronized with Microsoft Update. Azure Automation itself is therefore not involved in either the deployment or the installation of the patches.

Microsoft essentially points users to Azure Automation for OS updates. However, Update Management can be used to orchestrate all kinds of patches (e.g., in connection with WSUS). Because WSUS patching is managed by Group Policy, you can patch the server without a schedule in Azure Automation and simply use the service for reporting in Log Analytics. You do not necessarily have to distribute patches with Azure Automation.

More Convenient Updating with Azure Automation

Azure Automation offers significantly more convenience when it

comes to update management. If you open Azure Automation in the Azure portal, you will find all the systems on which update management is configured in the *Update management* section. You can also discover whether the update agent is ready on the monitored system. If important updates are missing, the status will be *Non-compliant*. This example uses *Windows Update* as the approval source and the *Automatic Updates* setting is for scheduled installations.

To schedule a deployment, click *Schedule update deployment* (Figures 3 and 4). The settings for a *New update deployment* with *Machines to update*, *Update classifications*, *Schedule settings*, *Maintenance window*, and so on are largely self-explanatory. In the *Reboot options* field you have to decide whether you want an automatic restart.

Instead of activating Update Management from within the VM, you can also actively add Azure VMs directly by selecting *Add Azure VMs*. To access the previously mentioned linked Log Analytics workspace, go to the *Related Resources | Linked workspace* section.

Last but not least, Azure Automation Update Management can provide the pre- and post-steps shown in Figure 2 as Azure Automation runbooks, which allows you to enforce the processing of these runbooks in the automation schedule. Scripts like this always run on the Azure platform and not on the VM. For example, you could use them in the context of automatically starting (and stopping) Azure VMs to update VMs that are normally switched off at the desired update time.

If you also want these automation runbooks to trigger actions on the VM, you need to have a *Hybrid Runbook Worker* running on the VM. Activating Update Management for the first time can take up to 15 minutes.

Runbooks

If you look closely at the VM, you will find the *Auto-shutdown* option in

the Operations section. This feature is also available when creating a new VM in the management window and helps newcomers in particular avoid unnecessary costs by shutting down VMs after 6pm, for example. The feature automatically takes advantage of process automation in Azure. Shutdown notifications are supported by web hooks or email.

A simple configuration option for automatically starting VMs is not available, although you can remedy this by using a runbook in Azure Automation in combination with a suitable schedule. To do so, go to the *Process Automation* section in your Automation account, then to *Runbooks*. When you get there, click *Create Runbook*. If you have ever dealt with PowerShell runbooks, or even Power Automate on the Microsoft Power Platform, you will feel at home here. Azure supports PowerShell, PowerShell Workflow, PowerShell (graphical), and Python runbooks.

If you don't want to start from scratch, Microsoft offers access to an extensive catalog of ready-to-run runbooks (under *Browse from Gallery*) from which you can draw inspiration. For example, the *Stop-Start-AzureVM (Scheduled VM Shutdown/Startup)* PowerShell workflow runbook connects to Microsoft Azure with Automation credentials and starts or stops a VM, a list of VMs, or all VMs in a subscription in parallel.

After clicking *Select*, the runbook appears in the Runbook editor with the *Runbooks | All* node. You can test the runbook by selecting *Test* then specifying your Azure SubscriptionID, VM, or list of desired VMs and the desired action (Start/Stop) as string parameters. *Start* then initiates the test run (Figure 5). However, you must first set the credentials under which the runbook will run in Azure Automation in the *Shared Resources | Credentials* section. This means you don't have to embed credentials in the source code.

Now you should be able to test the PowerShell workflow runbook for automatic VM start/stop, because

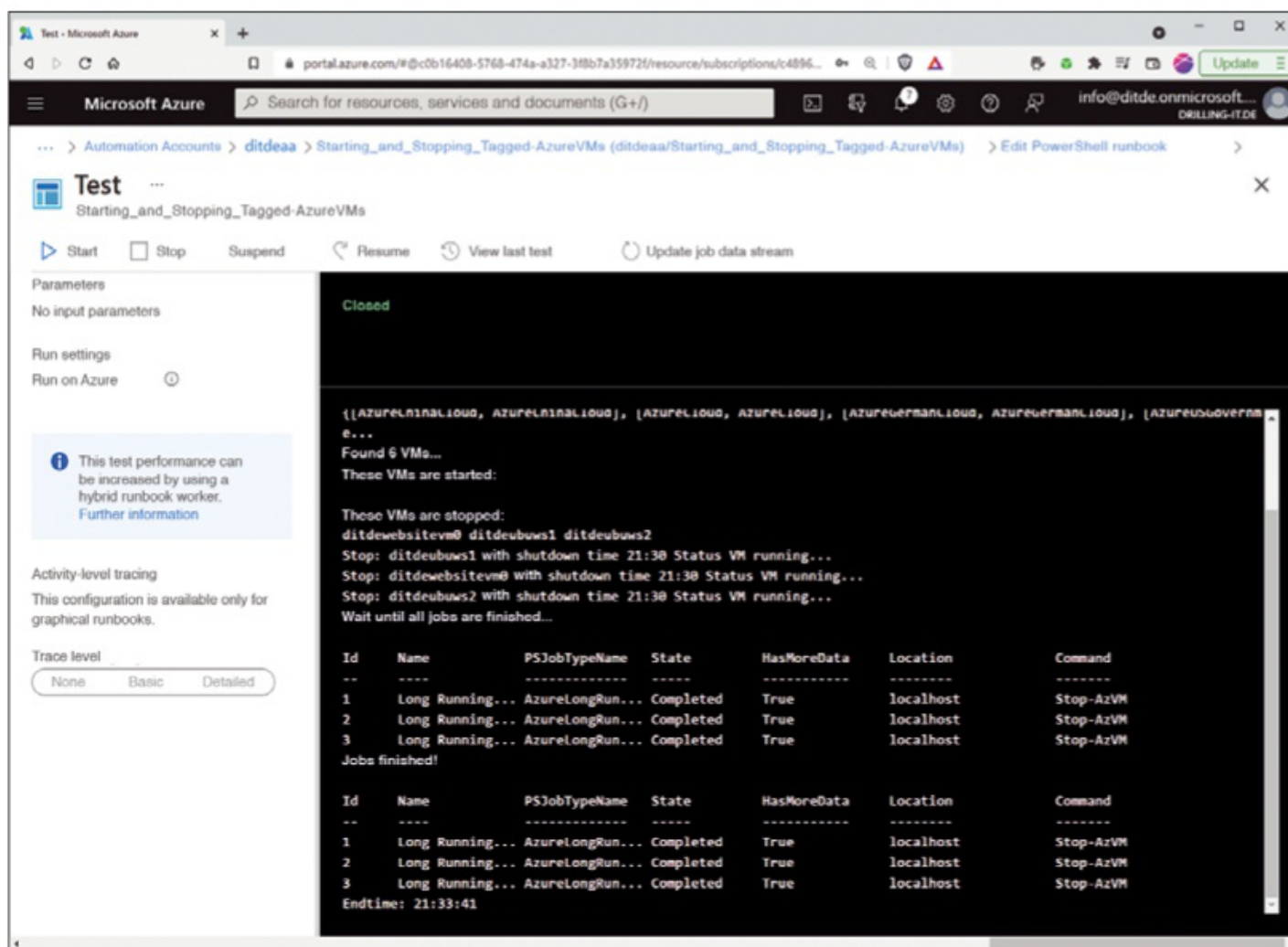


Figure 5: Successful test run of a PowerShell runbook.

the credentials are only referenced in the code:

```
$credential = Get-AutomationPSCredential 2
-Name 'democred1' Login-AzureRmAccount 2
-Credential $credential
```

Of course, PowerShell runbooks offer more flexibility. For example, you can search for the PowerShell *Stop/Start all or only tagged Azure VMs* runbook in the Runbook Catalog, import it, and store it under a name of your choice. The runbook is even compatible with PowerShell Core and connects with an Azure Run As account. It retrieves the power status of the VMs (started, stopped, de-allocated) and switches the runners off and the non-runners

on. The VMs in question can even have a tag for this.

As an alternative to the credentials used previously, you now also need to create a Run As account under the Account Settings section to be able to use it for authentication in the runbook. The Run As account then has its own application ID and a fingerprint, and the Run As connection appears in *Shared Resources | Connections*. Listing 1 shows how to use a service principal to manage the authentication of the runbook.

Again, you will want to test this runbook first. It expects an action (Start/Stop) and optionally the desired tag name and value as parameters. The script can be extended to suit your needs given

appropriate PowerShell skills.

If the test is successful, you can then publish the runbook and use it. Finally, to automate the process, all you need to do is add the runbook to the schedule after publishing. In Azure Automation you can do this in the *Shared resources | Schedules* section by selecting *Add a schedule*.

Of course, you can also start the runbook manually at any time and view the associated logs in the *Logs* tab if necessary.

Conclusions

All told, Azure Automation comprises several interacting toolsets. The service is very powerful and covers many aspects beyond the scope of this article. Other areas of automation include infrastructure provisioning or configuration management. Because a PowerShell or Python runbook can create or control almost any type of Azure resource, the sky is basically the limit. ■

Info

[1] Update Management overview:
[\[https://docs.microsoft.com/en-us/azure/automation/update-management/media/overview/update-mgmt-workflow.png\]](https://docs.microsoft.com/en-us/azure/automation/update-management/media/overview/update-mgmt-workflow.png)

The Author

Thomas Drilling has been a full-time freelance journalist and editor for science and IT magazines for more than 10 years. He and his team make contributions on the topics of open source, Linux, servers, IT administration, and Mac OS X. Drilling is also a book author and publisher; advises small and medium-sized enterprises as an IT consultant; and lectures on Linux, open source, and IT security.

Listing 1: Authenticating the Runbook

```

01 try
02 {
03     # Ensures you do not inherit an AzContext in your runbook
04     $null = Disable-AzContextAutosave -Scope Process
05     $Conn = Get-AutomationConnection -Name AzureRunAsConnection
06     $null = Connect-AzAccount -ServicePrincipal -Tenant $Conn.TenantID -ApplicationId
           $Conn.ApplicationID -CertificateThumbprint $Conn.CertificateThumbprint
07     Write-Output "Successfully logged into Azure."
```


CLEAR OFF YOUR BOOKSHELF WITH DIGITAL ARCHIVES

Complete your collection of *Linux Magazine* and *ADMIN Network & Security* with our Digital Archive Bundles.

You get a full year of issues in PDF format to access at any time from any device.



Lead Image © enki, 123RF.com

<https://bit.ly/archive-bundle>

2021
Archives
Available
Now!



Tracking down problems with Jaeger

Hunter

The various components of cloud-native applications are always exchanging information, which makes troubleshooting difficult. The Jaeger tracing framework helps hunt down the perpetrators. By Martin Loschwitz

Administrators facing a container-based setup with distributed applications for the first time in their career might hark back to the past and secretly think the old ways were better. From their perspective, at least, you might understand how this misconception comes about. People who used to be responsible for troubleshooting had a few fairly obvious starting points. Large monolithic programs such as MySQL simply output error messages. A look at the logfile was therefore often all it took to get at least a hint of where to look. If nothing useful could be found in the logfile, you still had the level below it as the starting point. For example, if communication between server and client did not work as described in the documentation, many an admin would turn to tools such as Tcpdump ([Figure 1](#)), which lets you read data traffic down to the lowest levels of a network connection for subsequent visualization with Wireshark to check for potential issues.

Also, the client could see potential errors and output appropriate messages on the terminal, if need be. Admins and developers can only dream of such simple debugging mechanisms in more modern applications. If you have ever experienced the frustration of tracking down problems in a distributed application, you will be fully aware of the complexity of this task. Realistically, the job can only be done if you are tackling a reasonably simple component with just a few microapplications. Application developers are therefore strongly advised to take a closer look at the Jaeger implementation of the Open Telemetry standard.

Modern Applications

As a reminder, the cloud-native architecture does have some advantages, such as implicit redundancy and the option to integrate dynamically external solutions such as Istio. On the downside, though, the complexity of

the individual application has grown exponentially. A direct comparison of the old and new worlds quickly illustrates this, and a database, as mentioned earlier, is an ideal candidate. Clients establish persistent connections to a database, establishing a connection once and then using it continuously until either one side officially terminates it or an error of some kind kills off all communication. In all of these cases, the server and client immediately notice that the other side can no longer be reached and acknowledge this with a clear-cut message.

Cloud-native distributed applications are totally thrown by this scenario; even the idea of the connection is alien to them. Cloud-native applications are built as microcomponents instead of large monoliths. In a cloud-ready environment, no single application handles all tasks. Instead, a number of small and highly specific applications designed for a single task is active.

Photo by Sebastian Molina fotografia on Unsplash


```

Black on White (bash)
192.168.2.135.62046: Flags [.], cksum 0xaf62 (correct), ack 989, win 84, options [nop,nop,TS val 3181164221
ecr 3895924554], length 0
15:08:50.126423 IP (tos 0x0, ttl 44, id 29769, offset 0, flags [DF], proto TCP (6), length 52)
17.57.12.16.443 > 192.168.2.135.62047: Flags [.], cksum 0xe6d5 (correct), ack 518, win 84, options [nop,n
op,TS val 3303457929 ecr 748623466], length 0
15:08:50.126425 IP (tos 0x0, ttl 44, id 31822, offset 0, flags [DF], proto TCP (6), length 337)
17.57.12.16.443 > 192.168.2.135.62046: Flags [P.], cksum 0xbdb2 (correct), seq 2979:3264, ack 989, win 84
, options [nop,nop,TS val 3181164221 ecr 3895924554], length 285
15:08:50.126504 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 52)
192.168.2.135.62046 > 17.57.12.16.443: Flags [.], cksum 0xa601 (correct), ack 3264, win 2043, options [no
p,nop,TS val 3895924711 ecr 3181164221], length 0
15:08:50.127488 IP (tos 0x0, ttl 64, id 0, offset 0, flags [DF], proto TCP (6), length 76)
192.168.2.135.62046 > 17.57.12.16.443: Flags [P.], cksum 0x567b (correct), seq 989:1013, ack 3264, win 20
48, options [nop,nop,TS val 3895924712 ecr 3181164221], length 24
15:08:50.127540 IP (tos 0x0, ttl 44, id 29770, offset 0, flags [DF], proto TCP (6), length 1492)
17.57.12.16.443 > 192.168.2.135.62047: Flags [.], cksum 0xd784 (correct), seq 1:1441, ack 518, win 84, op
tions [nop,nop,TS val 3303457929 ecr 748623466], length 1440
15:08:50.127547 IP (tos 0x0, ttl 44, id 29771, offset 0, flags [DF], proto TCP (6), length 1492)
17.57.12.16.443 > 192.168.2.135.62047: Flags [P.], cksum 0x253c (correct), seq 1441:2881, ack 518, win 84
, options [nop,nop,TS val 3303457929 ecr 748623466], length 1440

```

Figure 1: Tools such as Tcpdump used to be all you needed to examine network connections at the application level.

Several approaches compete for the role as the gold standard for communication between these components. RESTful APIs that are based on HTTP(S) are widely used today. Solutions such as a high-performance Remote Procedure Call (e.g., gRPC) framework also play a role. What they have in common is that they do not rely exclusively on stateful connections, like the database example.

As the number of microcomponents in an application increases, the number of potential communication interfaces increases exponentially. Recent cloud-native applications in particular are anything but frugal in terms of the number of microapplications they contain.

Many Apps

One microapplication then serves as a point of contact (for example, for communication with clients). A second microapplication in the background receives input forwarded from the first, evaluates it, and sends it to a third microapplication, which then stores the data somewhere on a disk.

A fourth application could monitor the content of the stored data and sound the alarm if certain content appears or certain events occur during a write. Microapplication five

could be used to deliver the alerts generated by the fourth component in the form of text messages by email, SMS, or a messenger service. What this relatively simple example already shows is that data can travel long distances on the wide network of a microservices architecture, undergoing regular transformations in a variety of ways and ending up with various recipients as fragments.

Complex Network

The end of the line is still far off in terms of complexity. In the past,

ADMIN regularly featured solutions such as Sidecar and Istio that do their best to expand on this chaos.

Istio, for example, comes with a Sidecar component that dynamically engages in the communication of microarchitecture components. While doing so, it handles a whole gamut of tasks: Istio can implement firewall rules, add SSL encryption to the communication endpoints on the fly, and distribute the incoming load for each of the application's individual components among the available instances (**Figure 2**).

From the client's point of view, it remains unclear with which

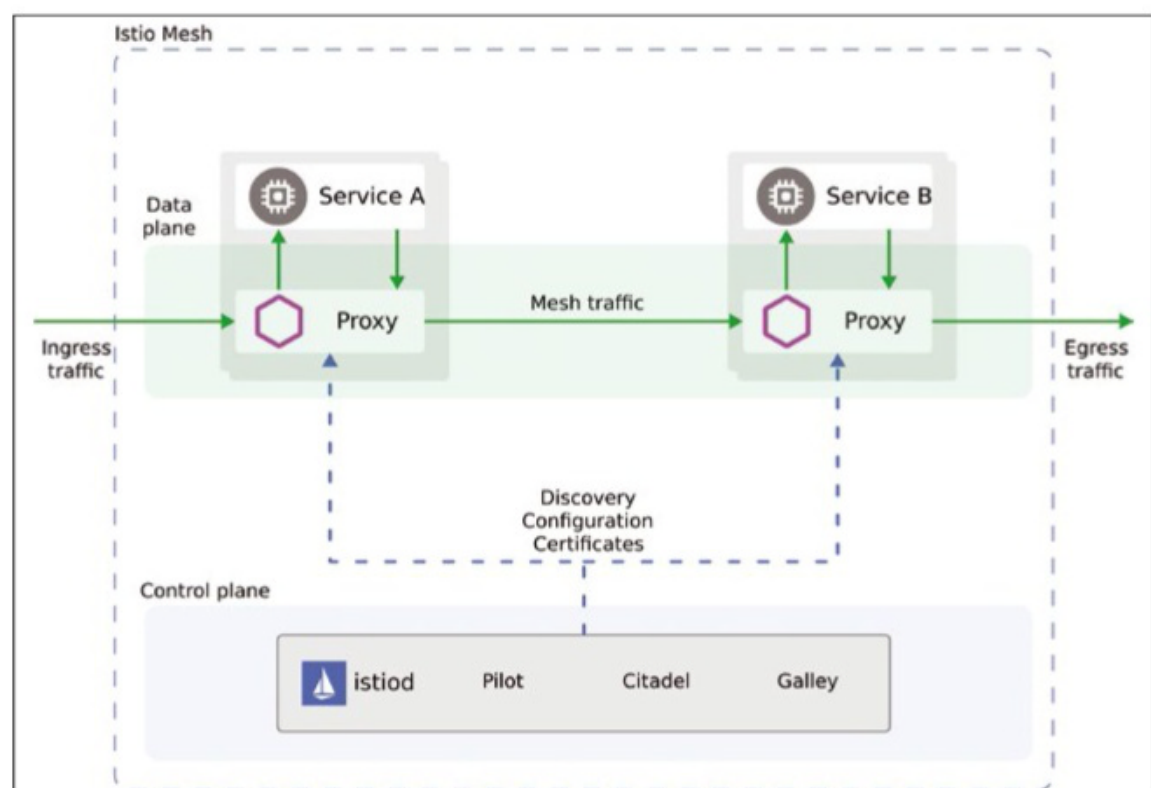


Figure 2: The architecture schematic for Istio shows that even a dynamically changing network will be so complex that tools like Tcpdump are of little use. © Istio

instance of an application's component it is currently communicating. Because most components in distributed applications also exchange data among one another, the majority of communication paths remain completely in the dark for the admin and developers of an application.

Case Debugging

The microarchitecture of an application shown here also makes it clear why debugging problems in this kind of environment can easily become a Herculean task. A very simple example makes this clear: Imagine a user who accesses a distributed application through a web application and wants to query the data stored there. Instead of the expected data, however, they only see an error message. So where does the problem lie?

The potential sources of error are almost infinite. The component that accepts the request from the web application or the load balancer upstream might not be working properly. Also, the load balancer between the first and second components might not be configured correctly and, as a result, simply drop the request. However, it could just as easily be because of network problems between the physical systems on which the application's components run – and these components are likely to be packaged in containers.

Other possibilities are that the request is reaching the database, but the database cannot respond appropriately or the persistent memory available to the database is faulty. Access to memory over the network might not be working because one of the switches in the setup has failed and is generating junk. The database might even be delivering the desired data, which then gets stuck somewhere on the way back.

How is an admin or developer, for which an active cloud-ready application already looks like the proverbial black box, supposed to find the root of the trouble now?

Conventional Means Will Fail

Classic approaches such as reading logfiles generated by individual parts of the application are often frustrating because the components do not generate any log messages – not because their developers have not bothered to implement the function correctly, but because, in a distributed system, individual containers often have no persistent storage for logs. Even if the app were capable of logging, it would not know where to put the logs, at least locally.

Classic monitoring approaches are also difficult or impossible to implement, because event monitoring or simply monitoring metrics data is of little practical help when you are debugging an individual problem. After all, you have nothing to gain if your monitoring says all systems are go because it is unable to detect individual problems. To cut a long story short, microarchitecture applications require a complete rethink of debugging and monitoring, which is where Open Telemetry and its Jaeger implementation come into play.

Standard and Application

Right from the outset, I need to clarify that Open Telemetry and Jaeger are not identical, although the terms are often used synonymously.

Open Telemetry is a standard [1] that describes a communication interface that application developers can integrate into their products to communicate over a defined protocol with the outside world. The goal is always to collect and export telemetry data (log and metrics data and tracing information from data streams) in a standardized format. On top of that, Open Telemetry now also offers a variety of clients for integration into programming languages, such as Go or Python.

Jaeger is a concrete implementation of tracing functions in line with the Open Telemetry standard that provides a framework that helps admins evaluate Open Telemetry data [2]. What sounds abstract and

complicated in theory is far easier to understand in practice. It is therefore worth recalling once again the fictitious microarchitecture application at the beginning of this article and the failure scenario described for it to illustrate the benefits of Open Telemetry and Jaeger.

Understanding Open Telemetry

If Open Telemetry is to be used, the developers of an application need to take this into account as early as the programming phase. Open Telemetry offers bindings for Go, for example, and Go is used in a large number of applications in today's cloud-ready universe. The first step on the way to effective tracing is to import the Open Telemetry bindings into the components of your own application.

Now is a good time to take a closer look at the terms used by Open Telemetry. The documentation for the standard in particular uses them so excessively that sooner or later you can no longer see the forest for the trees because of the large numbers of spans and traces. However, the topic is not as complex as the documentation makes it seem. The two basic terms, “spans” and “traces,” are a good way to illustrate the fundamentals of Open Telemetry.

Spans and Traces

In Open Telemetry-speak, a span is a set of data produced by running an arbitrary operation. If an application is prepared by a client library for Open Telemetry, each function generates a span with each call. The span has a unique ID (which includes the name of the function that was called), the exact time of the call, and information about the time taken to complete the task. Spans can be connected or nested within each other. Thus, if function A calls function B, two spans logically connected by nesting are created.

Traces, on the other hand, describe a group of spans that are directly causally related. You can probably already

see why Open Telemetry is practically indispensable for debugging microarchitecture apps. Traces are not limited to individual components of an application. If event 1 in application A causes event 2 to occur in application B, then several spans are created, depending on the application type and function, but each case only has one trace.

The practical benefits of Open Telemetry become apparent when you use a tracing framework to relate spans and traces visually, making it possible to see which applications called which functions where and when. You can also see the resulting events, the available data, and what happened to the data.

Expanding the Focus

Today, anyone looking at Open Telemetry and Jaeger for the first time will find significantly more functionality than was the case just a few months ago. The Open Telemetry developers have continuously expanded the scope of their standard. While the focus was originally on tracing information, today the standard also specifies formats for collecting metrics data and logfiles.

Tracing frameworks based on the Open Telemetry standard can therefore be used to expand the database for debugging. Having the traces available, as well as the corresponding data processing error messages logged by the application, helps to simplify troubleshooting further. Open Telemetry also jumps into the monitoring and logging gap for microarchitecture apps described earlier. For both types of data, the standard comes with formats that allow applications to generate suitable data. The tracing frameworks needed for analysis have also grown in number and become more diverse. Some of the data generated by Open Telemetry can be processed, for example, by Prometheus (metrics data) or Loki (log data).

Jaeger as a Framework

In the example here, after integrating the Open Telemetry framework, the admin or developer has an application that generates metrics, logging, and tracing data. However, that is only half the battle because you also need to display and make sense of the data. In this case, Jaeger becomes a

possible implementation of a tracing framework. Although Jaeger initially offered client libraries, it has since retired them in favor of Open Telemetry's counterparts.

What is left is the Jaeger server, which itself comprises several components. Interestingly, they do not just run on a single host: A Jaeger agent supports clients on the target systems (or in the target containers), which often exist in large numbers. The task is as simple as can be: to field the data created and collected by the clients. In container applications, the agent is usually integrated as another Sidecar. In practice, it ensures that the application itself does not need to know where to send the acquired trace, log, and metrics data.

What seems to be trivial is of great importance in practice, because the Jaeger configuration is autonomous and independent of the application's own configuration thanks to the Jaeger agent, which means it can also be changed on the fly.

From the App to the User

What happens to the data once it reaches the Jaeger agent depends on

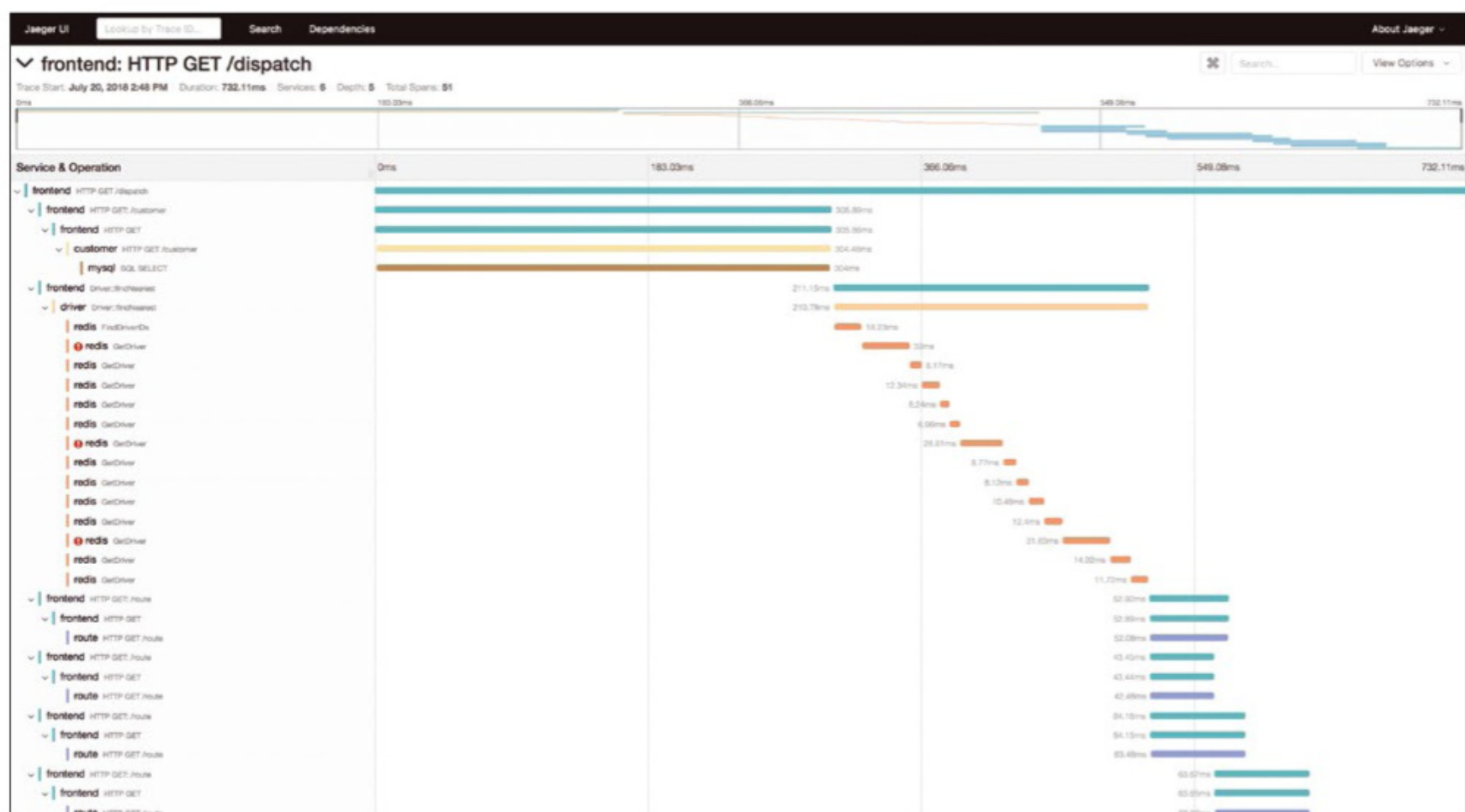


Figure 3: Jaeger works with spans, which it groups into traces, making it possible to trace the history of data and of events that process that data accurately. © Jaeger

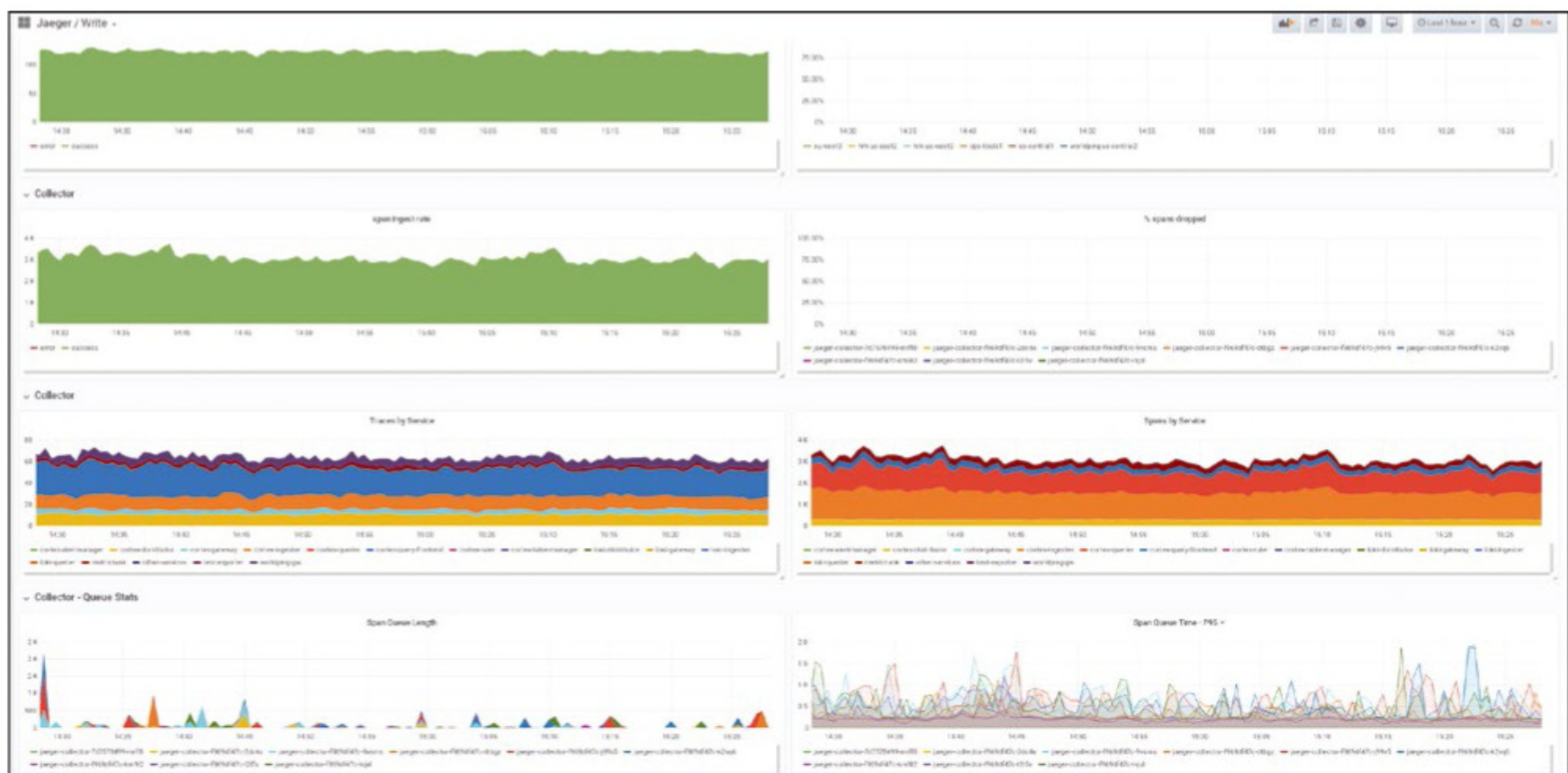


Figure 4: Because Jaeger now also has an interface to output data in Prometheus format, metric data acquired by Jaeger can be displayed directly in Prometheus and therefore also in Grafana. © Grafana

the setup. Several options initially rely on the Jaeger collector, which in turn collects and samples all the available data from the various agent instances. Sampling is primarily intended to remove redundant information from the acquired data to reduce the overall volume. With no rigid rules, the collector uses adaptive sampling. The administrator or the developer can influence this process in a separate configuration, if needed.

From the collector, the data then moves on to a database. Jaeger relies on persistent storage in the background to save and process the

acquired information. However, the choices here are Elasticsearch, Cassandra, or Kafka instead of legacy relational databases. Once the data is stored, the developers recommend using Apache Spark to optimize the database content with Spark jobs available in the Jaeger repository. Finally, the Jaeger query component reads data from the traces database on the basis of user-defined parameters. All visualization tools, including Jaeger's own, always access the query component and never access the database directly. The database obviously will be under a tremendous load with new traces arriving

on one side and queries for existing data piling in from the query component on the other. Jaeger query, by the way, is the application that uses Jaeger's own user interface (UI; [Figure 3](#)) when it queries data from Jaeger.

To Cache or Not To Cache?

Because of the potentially high base load, the Jaeger developers have set up an option for caching the database with a (possibly additional) Kafka instance. The collector then writes its data to the cache instead of directly to the main database,

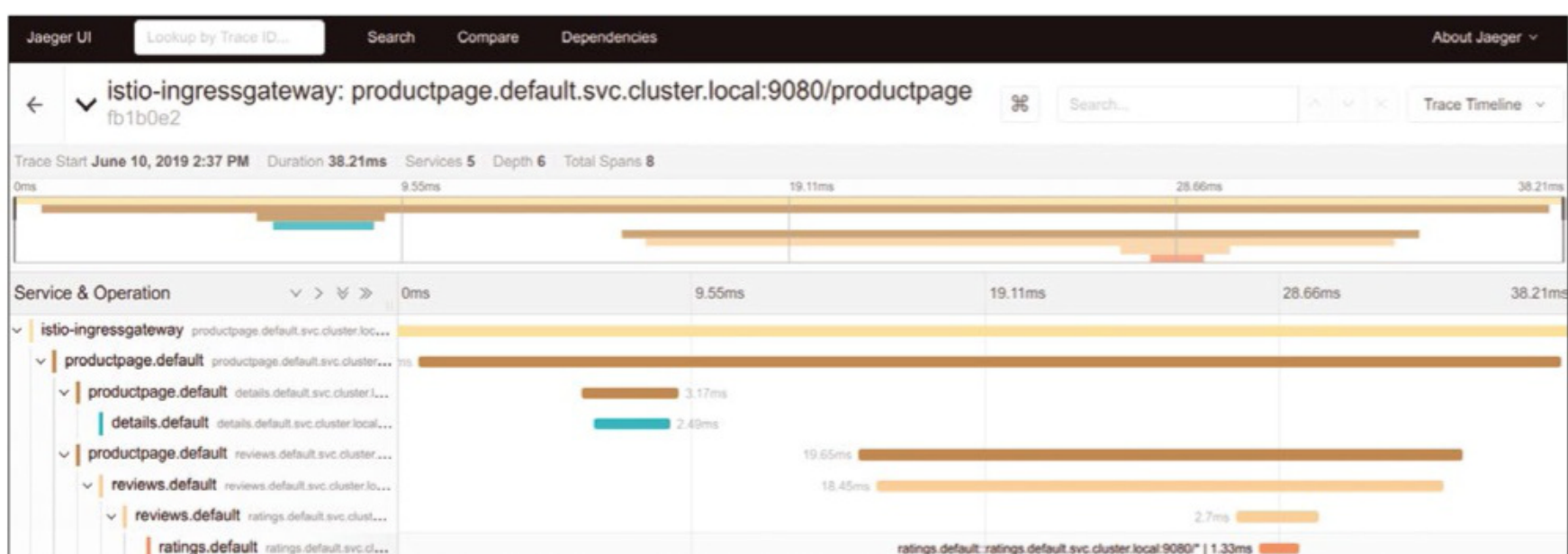


Figure 5: Istio has a native Jaeger connection, so actions initiated by, say, the Istio ingress gateway can be displayed as traces in Jaeger without any further action. © Istio

and the intermediary Kafka instance asynchronously pushes the data to the persistent database in the background.

In this setup, the additional load produced by Spark no longer directly affects the back-end database but is offloaded to the caching Kafka instance by Apache Flink. This relatively complex architecture does have practical uses. Large microarchitecture applications often produce large volumes of traces with millions of spans in a very short time. The interaction of the tools mentioned here easily keeps the setup usable and stable in this case.

Jaeger in Practice

In everyday life, Jaeger is just as relevant for administrators as it is for application developers. Admins use the tool primarily for troubleshooting, as described, whereas developers also use Jaeger to find out whether changes to the individual components of a microservice application deliver the desired results. However, how can you ideally set up your Jaeger instance to leverage it to the full extent in a fast and efficient way? How can Jaeger be combined usefully with other tools such as Prometheus and Grafana for more benefits?

The good news is that ready-made how-tos exist that describe Jaeger

integration with the popular applications of today's containerized world. Moreover, many tools, such as Prometheus, have decided to support Jaeger integration right out of the box.

Jaeger and Prometheus are often mistakenly considered competitors because they both have something to do with metrics data and monitoring. In fact, however, the two can combine perfectly to help you boost the efficiency of platform monitoring. Jaeger's query component, for example, now has a `/metrics` endpoint through which it outputs acquired metrics data in the Prometheus format (Figure 4). In doing so, the component acts as a kind of exporter for Prometheus data.

Once the data from Jaeger arrives in Prometheus, you can do anything the tool allows – usually in combination with Grafana. Visualization, for example, then becomes a breeze. Although the Jaeger UI is optimized as a standalone component for displaying traces and spans, Grafana in combination with Prometheus impresses when it comes to processing metrics data.

Istio and Envoy are further outstanding examples of all-around successful Jaeger integration. The Istio service mesh now also integrates Open Telemetry functionality so that Istio operations can be displayed in detail in Jaeger (Figure 5). The Envoy load

balancer component included in Istio can also be excellently linked with Jaeger (Figure 6). More detailed information regarding the steps required for integration can be found in the respective manuals.

Conclusions

If programmers fail to integrate Jaeger at the development level, a Jaeger server instance – no matter how well it is set up – will be of no use. Conversely, you need to plan for a working Jaeger framework in your data centers from the outset. Practice shows that getting a Jaeger setup up and running is anything but trivial. System administrators also tend to underestimate the load that a setup like this must shoulder. It can be substantial if several microservices generate traces at the same time; therefore, the hardware for the setup must be dimensioned appropriately. ■

Info

[1] Open Telemetry standard:

[<https://opentelemetry.io>]

[2] Jaeger tracing:

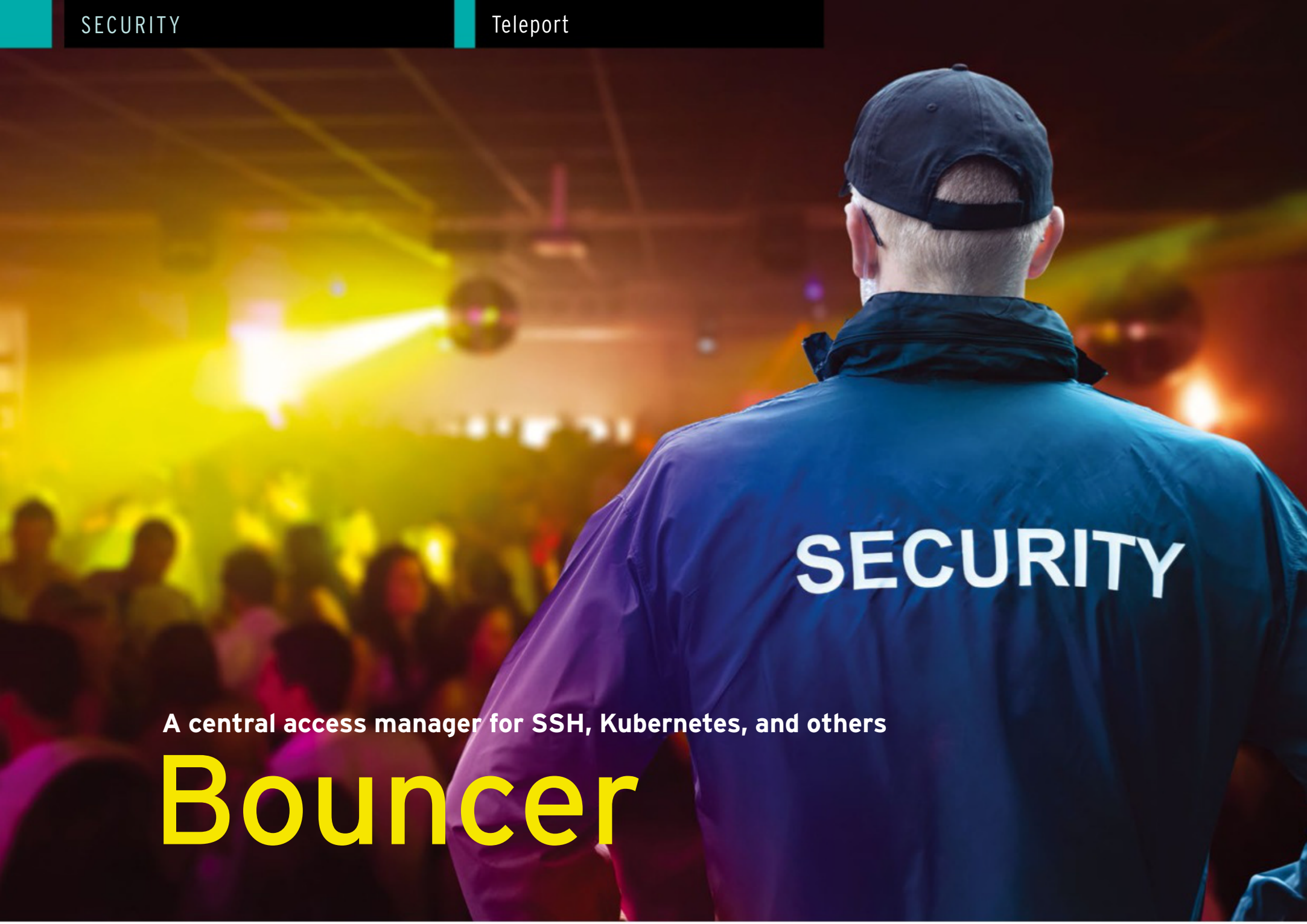
[<https://www.jaegertracing.io>]

The Author

Freelance journalist Martin Gerhard Loschwitz focuses primarily on topics such as OpenStack, Kubernetes, and Ceph.



Figure 6: The Istio Envoy component links up well with Jaeger to represent and track traffic flows visually. © Arvind Thangamani



A central access manager for SSH, Kubernetes, and others

Bouncer

Teleport centrally manages logins against various protocols, including SSH, Kubernetes, and databases. Functions such as two-factor authentication are included in the scope of delivery, as is management of your own certificates. By Martin Loschwitz

A decade and a half ago, when security and compliance were not as dominant in some places as they are today, the number of accounts and passwords was something like manageable and many an admin got rid of local accounts altogether in favor of root on their SSH-only machines. Instead, a password for root worked everywhere in the setup and – with a little luck – was encrypted for storage in a central password store somewhere.

Today, however, this practice is completely unthinkable. Various security standards (e.g., Payment Card Industry Data Security Standard, PCI DSS) now stipulate that it must be possible to trace who changed what on which systems and when, which makes individual accounts mandatory. Today, even the toughest deniers of the need for compliance tend to avoid root logins over secure shell (SSH) with just password protection. Most

distributions even prohibit this practice in the default configuration. Sudo and SSH keys for individual access are the means of choice instead. This practice only makes sense, especially in setups where admins are really only entrusted with the operation and maintenance of machines that can be accessed by SSH. However, this is decreasingly the case today. DevOps, the cloud, and containers have made information technology (IT) far more heterogeneous. Kubernetes APIs and databases need credentials, along with various other services, and all of them use their own protocols for authentication. Even SSH is not quite as clear-cut as it seems at first: In the interest of security, it is quite common today not to run systems with a direct connection to the Internet if they do not need the access. SUSE Manager, Red Hat Satellite, and the like have long since found a solution to the problem

of delivering updates and other essential features to systems without a direct Internet connection. In return, however, such systems can no longer be accessed directly over SSH. Instead, a jump host or cluster workstation is used as a central access point from which the admin can shift onto the target system. To do this, however, you need to remember the sequences of hosts that take you to the destination.

Teleport

One way to approach this problem is with Teleport, which promises to consolidate “connectivity, authentication, authorization, and audit into a single platform to improve security and agility” [1]. The developers do this by eliminating nested SSH jump hosts, standardizing SSH authentication within a setup by issuing and revoking its own Secure Sockets Layer (SSL)

certificate authority (CA) and X.509 certificates (instead of relying on simple SSH keys), and adding two-factor authentication (2FA), which is difficult to do in SSH with on-board tools. The real killer feature of the software, though, is the use of SSH as a tunneling protocol to connect you to services such as Kubernetes, popular databases, and other services, thus converting the SSH server into a multitasking connectivity wizard, which in turn promises to take worries relating to authentication and authorization off your shoulders. In this article, I look at whether these claims pan out and examine the features of Teleport.

Architecture

For a deeper understanding, it is extremely useful to take a look at the Teleport architecture ([Figure 1](#)). Under the hood, a Teleport instance comprises three self-sufficient services that are part of the same binary. Teleport comes as a single large Go file. Although the individual parts of Teleport support cluster mode, it is not technically necessary.

Initially, the Teleport proxy plays a significant role.

It listens on the outside for incoming connections and acts as the SSH server (i.e., it speaks the SSH protocol). When it receives an incoming connection from a client, it first communicates in the background with the authentication component, usually abbreviated Auth. The question of whether a client is granted access to a target system is decided there: The Auth component is also a full SSL CA.

If a client logs in to the Auth component directly with a valid X.509 key issued by the Auth component itself, the connection to the target system enters the next phase. If the client uses a combination of a password and username instead, the Auth component first creates a valid X.509 certificate with restricted validity (usually 12 hours) and sends it back to the client. The second phase of establishing the connection then follows. Teleport is used in node mode in this case. On each node of a Teleport cluster, Teleport also runs as an SSH server in node mode. Once the client has a valid X.509 certificate, it establishes the connection to the target system via the Teleport proxy. A shell is then available to the client here, just as it would be in setups without Teleport.

Experienced admins who are always reluctant to accept fundamental innovations may turn up their noses at this point. Replacing OpenSSH, which many admins know and appreciate, is an invasive intervention in terms of a system's overall architecture, and Teleport doesn't offer that much added value at first glance. After all, you can also use X.509 certificates with plain vanilla SSH. To understand

the practical relevance of Teleport, it is worth taking a closer look at the various features that Teleport brings to setups.

Doing More

At the top is Teleport's ability to handle SSL certificates. OpenSSH can also check and evaluate certificates sent by clients on the basis of an existing SSL CA; however, the entire handling of the CA in such a setup is the administrator's responsibility, and any admin who has ever had an SSL CA on their plate can judge the complexity of this undertaking from their own experience.

Teleport at least takes some of the load off the admin's shoulders here, but it does even more: Because Teleport automates certificate handling, it also contributes to security by limiting the validity of any certificate. The certificates issued by Teleport are valid for a maximum of 12 hours, so even if they fall into the wrong hands, the damage can be contained.

Additionally, Teleport largely hides the entire certificate handling process from the user. Admins only have to worry about SSL during the initial

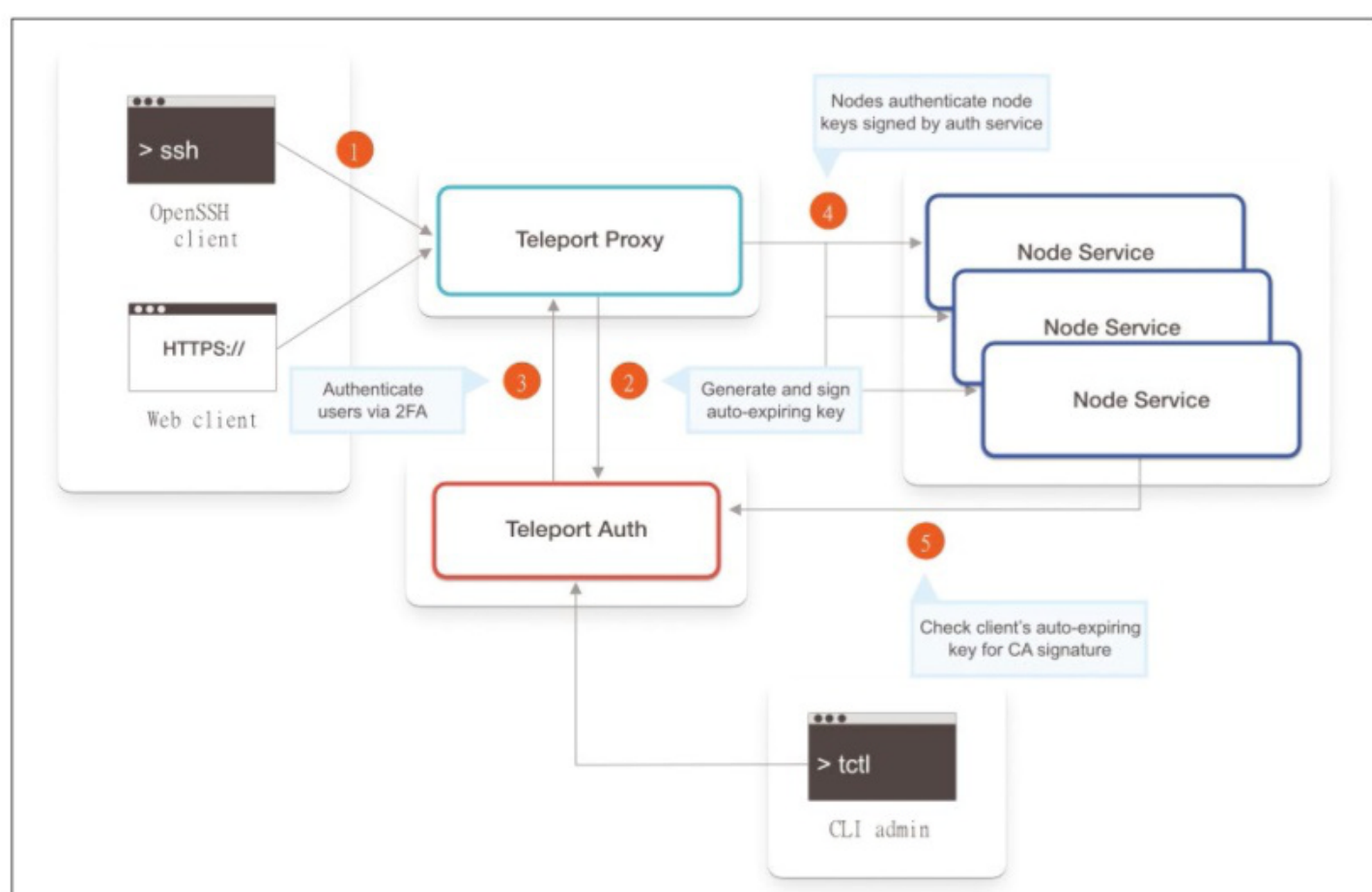


Figure 1: At its core, Teleport consists of three components: the proxy, the authentication component, and the node component. All nodes running a Teleport node together form a cluster as a logical unit.

setup. Once Teleport is up and running, everything works on its own, which makes everyday administrative work far easier.

2FA

The vast majority of setups that rely on SSH do not provide support for a second factor. The classic setup with SSH keys only handles 2FA with some add-ons, whereas it is an essential part of the entire authentication process in Teleport.

Teleport hides the X.509 certificates from users because they don't need to be familiar with the technical details of certificate handling to use Teleport. The Teleport shell is a separate client for user logins with their usernames, passwords, and a second factor. The SSL certificate for the client is then issued in the background. Teleport is far more secure out of the box than standard key-based SSH.

Access to Other Services

Another unique selling point for Teleport is that it can run both as an SSH server and as a gateway for any protocol in any direction. SSH mainly benefits.

The example I referred to earlier of connections across multiple jump hosts is very easy to implement in Teleport. In the background, Teleport keeps its own list of machines that belong to the overall setup. The

developers call this a cluster. Each machine running the Teleport Node service (the Teleport daemon with the node role) belongs to the same cluster. The SSL certificates Teleport issues are valid throughout the cluster; once a user has been authenticated on their own client, the certificate indirectly gives them access to all available resources.

In terms of node discovery, Teleport uses several mechanisms. Teleport nodes can be discovered by the classic domain name service (DNS). However, this is not the last word, because Teleport comes with its own discovery protocol (**Figure 2**). Under the hood, it is far more complex than you might expect.

The proxy servers I mentioned earlier play an important role here because multiple instances are possible – high availability is an essential part of the Teleport strategy. Proxy servers from the same cluster communicate with each other and exchange information about the target systems they can see. In this case, “see” means that the Teleport node component is active on a node: If you install the node component on a target system, you are basically running a number of commands to add the system to the existing cluster and specifying at least one of the existing proxy servers as the target for the join command.

When done, any target system can talk to any proxy server; in turn, the Teleport proxy servers can be nested

conveniently. Proxy 1 then receives information about all visible nodes from Leaf proxy 1 and shares it with the other existing Teleport proxies. A client that wants to connect to a target system therefore only needs to know its IP address or hostname. The user, on the other hand, no longer needs to remember which path they have to take through a mess of nested networks to reach their destination. This path is set up dynamically between the Teleport client and the proxy to which the client connects in the first step.

This arrangement is very practical, especially in the context of high availability. If you install a load balancer upstream of your Teleport proxies, the client only has to remember one address and can be certain of reaching any other host in the setup from any of the proxies running behind it. Modern setups that are broken down into multiple network segments and demilitarized zones (DMZs; subnetworks that contain and expose external-facing services to the Internet) and that implement extensive security measures see a huge convenience gain.

Security and Compliance

Built-in certificate management and the solution of the jump host problem are sufficient added value – compared with commercially available solutions – to justify the use of Teleport.

However, the benefits Teleport offers are still far from exhausted. Importantly, the Teleport developers have given a great deal of thought to the issues of security and compliance. If you operate Teleport in an environment that handles highly sensitive data, you might also be subject to strict legislation. In

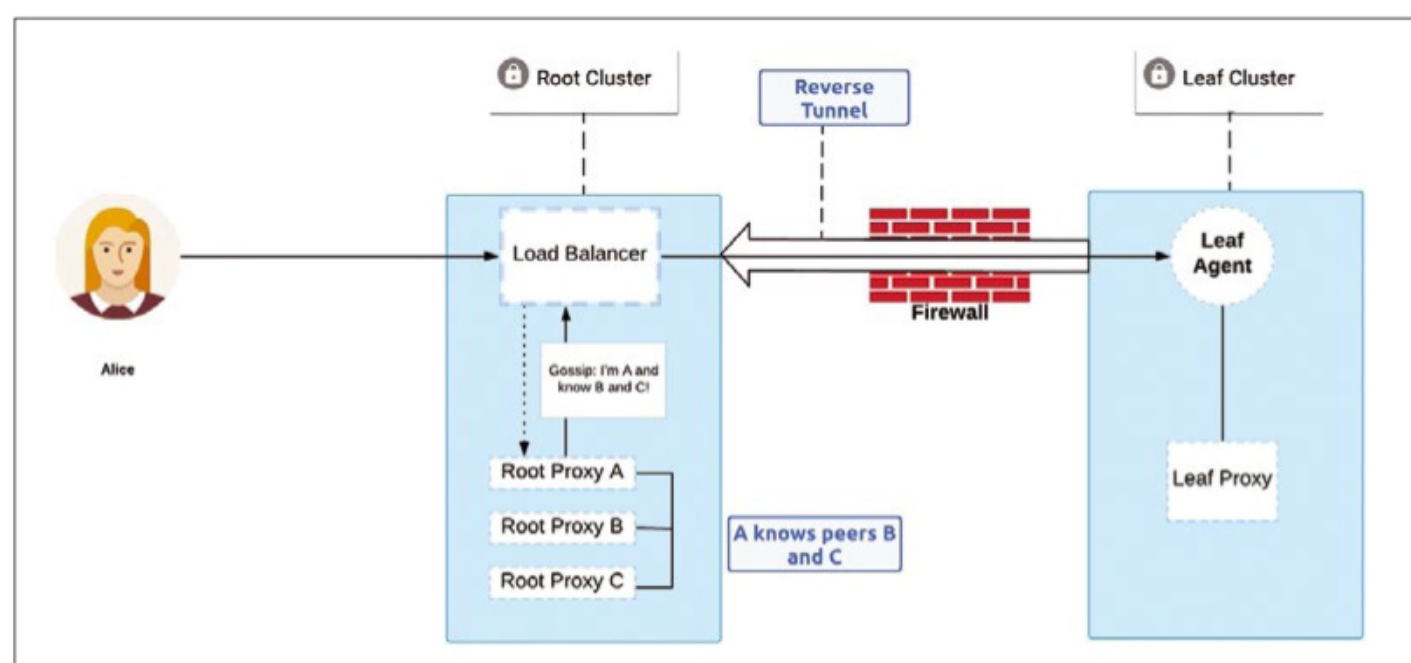
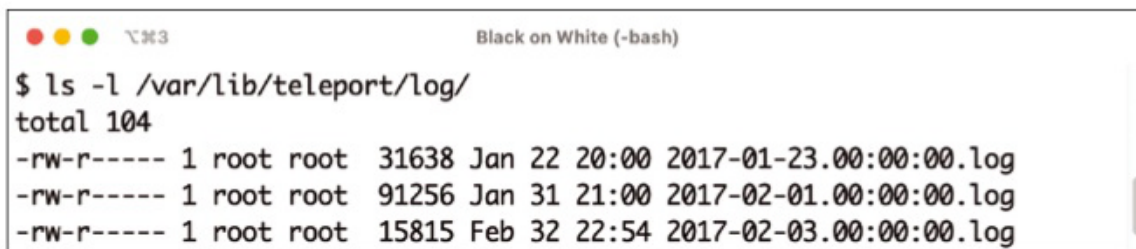


Figure 2: Teleport comes with its own node discovery protocol at the proxy level, making complex jump host setups a thing of the past. © Teleport



```

$ ls -l /var/lib/teleport/log/
total 104
-rw-r----- 1 root root 31638 Jan 22 20:00 2017-01-23.00:00:00.log
-rw-r----- 1 root root 91256 Jan 31 21:00 2017-02-01.00:00:00.log
-rw-r----- 1 root root 15815 Feb 32 22:54 2017-02-03.00:00:00.log

```

Figure 3: Teleport logs the commands and traffic it sees in several ways – Audit Logs are the most basic form of logging.

Teleport's home market, this legislation includes the Health Insurance Portability and Accountability Act (HIPAA), which defines strict standards for handling health data in the United States. Anyone who has squared up to compliance legislation will find many specifications and guidelines in HIPAA that are also found in relevant regulations in Europe, above all in the General Data Protection Regulation (GDPR). Teleport is officially certified under the HIPAA rules. But in Europe, certification according to the rules of the System and Organization Controls (SOC) 2 guideline is available and likely far more important because it does not stick to US-specific laws but is an accepted standard in the industry for applications related to security-relevant data. In other words, deploying Teleport in your environment if you are in Germany will make it easier for you to describe it within the framework of the Federal Office for Information Security (BSI) certification, because a matching report for this certification is readily available from the Teleport developers.

eBPF On Board

Another Teleport feature, which the documentation very unjustly just touches on, is worthy of mention. Teleport offers comprehensive auditing and logging capabilities for the systems on which it runs, a feature you will not often find in comparable solutions.

Teleport handles tasks that are specific to the architecture of the solution. If a component is responsible for centralized login management, it is easy for this component to keep records of logins. Teleport refers to this

as the Audit Log. The solution records who logged on to which server in the Teleport cluster at any given time and where the connection originated (Figure 3). The service then provides the information as a structured JSON file, which can be easily processed downstream in various services outside of Teleport.

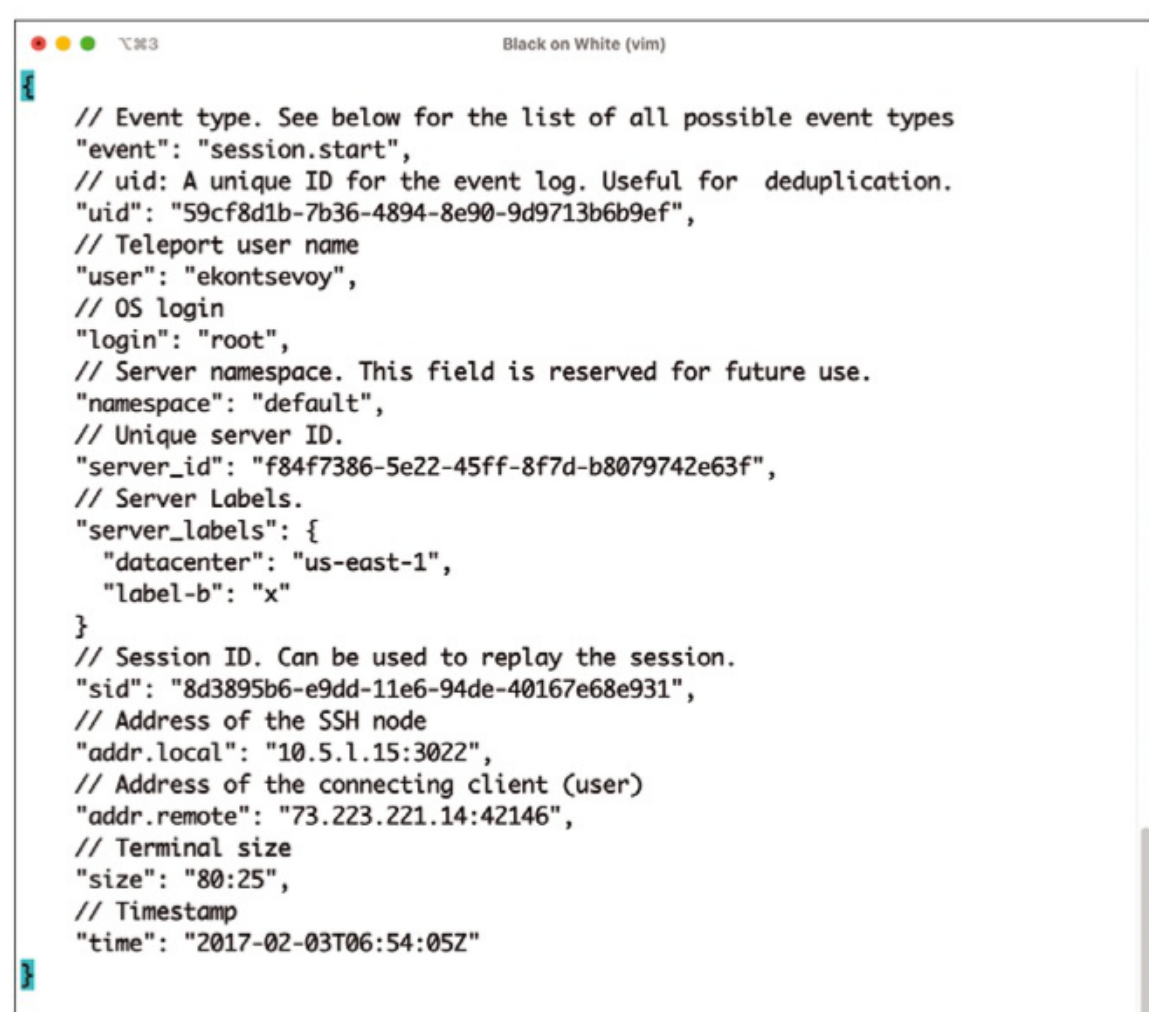
Of course, log data of this kind also needs to be available externally.

A successful attack could leave you with nothing but manipulated logs. With the use of an external data source, you might at least be able to determine the time of the hack to facilitate forensics. However, the market offers virtually nothing in the line of ready-made integration for logfiles originating from Teleport, such as for Loki or

the Elasticsearch-Logstash-Kibana (ELK) stack, which forces you into a do-it-yourself integration of the Teleport Audit Logs with a choice of logging system, if this functionality is needed.

However, Teleport's logging capabilities go far beyond recording the meta-data for connections (Figure 4). The developers have taken advantage of a technique offered by the Linux kernel itself – one that is brilliant but still attracts too little attention in the free/libre open source software (F/LOSS) world: the extended Berkeley Packet Filter (eBPF).

eBPF has been covered in our sister publication *Linux Magazine* [2] on several occasions. In practical terms, eBPFs are small virtual machines written in Rust that can be loaded into the running Linux kernel. Once there, they can implement arbitrary functions over a defined API. If you set this up correctly, eBPF can be given complete access to the traffic flow of a system or individual applications. eBPF is significantly more powerful than comparable nftables-based packet filters.



```

// Event type. See below for the list of all possible event types
"event": "session.start",
// uid: A unique ID for the event log. Useful for deduplication.
"uid": "59cf8d1b-7b36-4894-8e90-9d9713b6b9ef",
// Teleport user name
"user": "ekontsevov",
// OS login
"login": "root",
// Server namespace. This field is reserved for future use.
"namespace": "default",
// Unique server ID.
"server_id": "f84f7386-5e22-45ff-8f7d-b8079742e63f",
// Server Labels.
"server_labels": {
  "datacenter": "us-east-1",
  "label-b": "x"
}
// Session ID. Can be used to replay the session.
"sid": "8d3895b6-e9dd-11e6-94de-40167e68e931",
// Address of the SSH node
"addr.local": "10.5.1.15:3022",
// Address of the connecting client (user)
"addr.remote": "73.223.221.14:42146",
// Terminal size
"size": "80:25",
// Timestamp
"time": "2017-02-03T06:54:05Z"

```

Figure 4: Checking the logs, you will find structured logfiles in JSON format, which can be easily processed on other systems. Currently, however, native integration with Loki or ELK is missing.

Increasing numbers of applications that rely on eBPF are making their way onto the market. Google has presented a framework in the form of Kernel Runtime Security Instrumentation (KRSI). The framework monitors complete data streams and sets up checkpoints at neuralgic points to enable applications to interrupt traffic flows. KRSI does this by combining eBPF, which implements the monitoring functionality, with the Linux Security Framework; applications such as SELinux and AppArmor use this in a similar way. Teleport can be made to terminate or interrupt ongoing sessions automatically if the content of the transmitted information meets certain criteria according to predefined benchmarks.

Preventing Data Leaks

A real-world example makes it clear that once an attacker has gained access to a system, no one can prevent them (if they are root) from arbitrarily offloading data from that system. Publicly defacing websites is no longer the focus for most hackers; rather the real target of attack is databases containing credit card data and passwords. From the point of view of the attacked company, uncontrolled and unwanted data leaks are a far bigger problem that, depending on the country, can have serious legal consequences.

In these cases, dynamic interruption of Teleport comes in handy, at least assuming the Teleport proxy nodes

themselves do not fall victim to an attack. With the Restricted Session feature, you can define keywords that must not appear in the transferred sessions and that will automatically cause Teleport to terminate the connection. These keywords can be configured with the use of Teleport's own API. The feature is dynamic; for example, you could store a keyword in a hidden row in a database and tell Teleport to jump to that row during the transfer. This example is rudimentary, of course – anyone who wants to exploit the capabilities of Teleport and eBPF will have no alternative but to implement a holistic security setup and get involved in a little development work. It is remarkable, however, that Teleport allows this kind of setup at all; nothing comparable could realistically be implemented with OpenSSH and nftables.

Teleport's logging features also take effect in a far less invasive mode. If you “only” want an audit trail, Teleport can log all the traffic from SSH sessions. In terms of accountability, some certifications now go so far as to require more than just individual accounts for individual users; they also want to have proof of who executed what commands on a system and at what time.

For the principle to work, of course, it is important to have the Teleport proxies operated by different people from the target systems, because bad guys could otherwise easily bypass the self-installed protection mechanisms. However, given this

clear demarcation of responsibilities, Teleport is ideally suited to meet the comprehensive audit requirements of various certifications.

If you intend to use Teleport in this way in Europe, you should be aware of the need to comply with data protection requirements under labor law. This is a little easier if you do not log and monitor the communication content and instead restrict it to allowed hosts. This process also works with eBPF. If the filter notices that packets are designated for an IP that has not been cleared, it automatically interrupts the connection on an application-specific basis (Figure 5).

Kubernetes? No problem!

The benefits of Teleport described so far are primarily under the hood, but Teleport also offers many more visible features. The most prominent of these is undoubtedly Teleport's ability to act as a tunnel agent not only for SSH, but also for a variety of other protocols. It's true that SSH was the nucleus, and the Teleport documentation clearly states in many places that Teleport is an SSH server at its core. In the meantime, however, the solution has grown to see itself more as a kind of mediator between the worlds, as a few examples will illustrate.

In addition to the SSH protocol, the Teleport proxy supports the API of the Kubernetes cluster fleet manager. After a user's initial login against the Teleport proxy (`tsh login`), `tsh` builds the user's environment variables so that any requests to, say, Kubernetes or PostgreSQL are automatically redirected through the proxy server to the target system. On the system where the admin runs `ssh user@host` to access their target system after logging in to the Teleport shell, commands for connecting to PostgreSQL could subsequently be executed with `kubectl` or `psql`. Teleport is even available for remote desktop protocol (RDP)-based access to desktop systems (Figure 6). From the user's point of view, this diversity further increases the

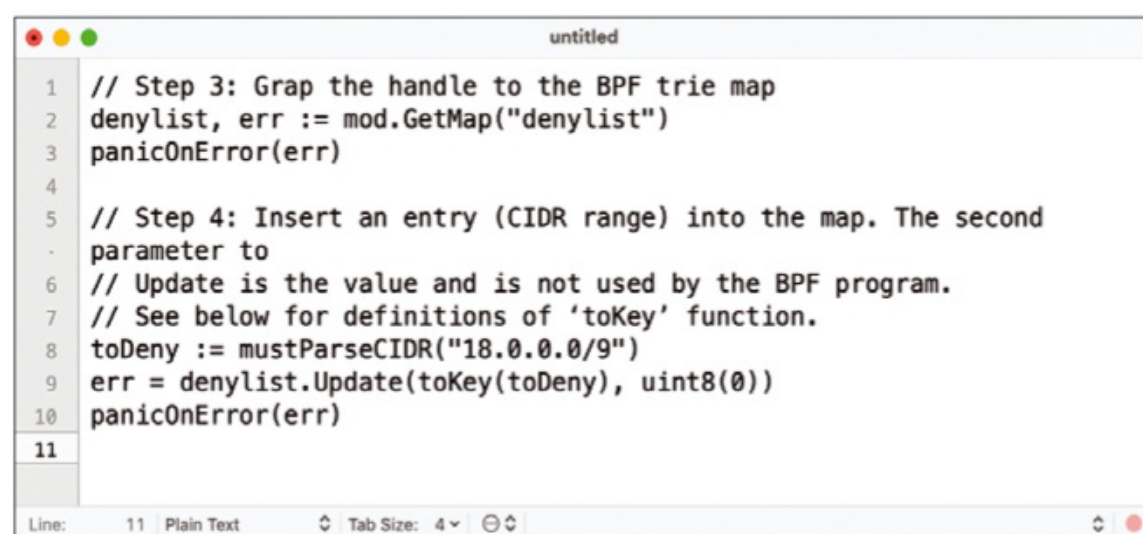


Figure 5: eBPF sample code used like this in Teleport takes down a connection if it sees data migrating to undesirable IP networks (`denylist`). © Teleport

convenience of using the solution: Where you previously had to worry about correctly setting up authentication for each individual application, Teleport simply takes this work off your hands. Anyone who has tried to manage the credentials for accessing a MySQL database in a meaningful way will eventually bundle their `.myrc` off into Ansible or some other software provisioning, configuration management, or application deployment tool and be annoyed about having to do so. However, this setup is anything but dynamic and requires regular adjustments if access credentials change.

By configuring the database to accept user certificates from the Teleport CA, issuing passwords is practically a thing of the past and is very likely to please compliance officers, too.

Conclusions

Teleport turns out to be a multitasked tool for establishing and managing connections between clients and their server-based services. The consistent focus on X.509 certificates means the end of passwords, as

Teleport puts the fun factor back into certificate management. Admins and users remain largely insulated from the complexity of a CA. Teleport handles this part almost completely autonomously – at least after the initial setup. Moreover, the features in Teleport that offer additional functionality are impressive – eBPF integration with some sample applications for KRSI being just one example. If you are looking to replace a mess of SSH, jump hosts, and half-baked security with a decent solution, Teleport is the way to go. This is all the more true because Teleport can also be used with a variety of

external authentication options. If so desired, users can even do without a separate directory service in their Teleport setups. ■

Info

[1] Teleport: [\[https://goteleport.com\]](https://goteleport.com)

[2] eBPF: [\[https://www.linuxpromagazine.com/index.php/Issues/2019/225/Getting-Insights-with-eBPF/\]](https://www.linuxpromagazine.com/index.php/Issues/2019/225/Getting-Insights-with-eBPF/)

The Author

Freelance journalist Martin Gerhard Loschwitz focuses primarily on topics such as OpenStack, Kubernetes, and Ceph.

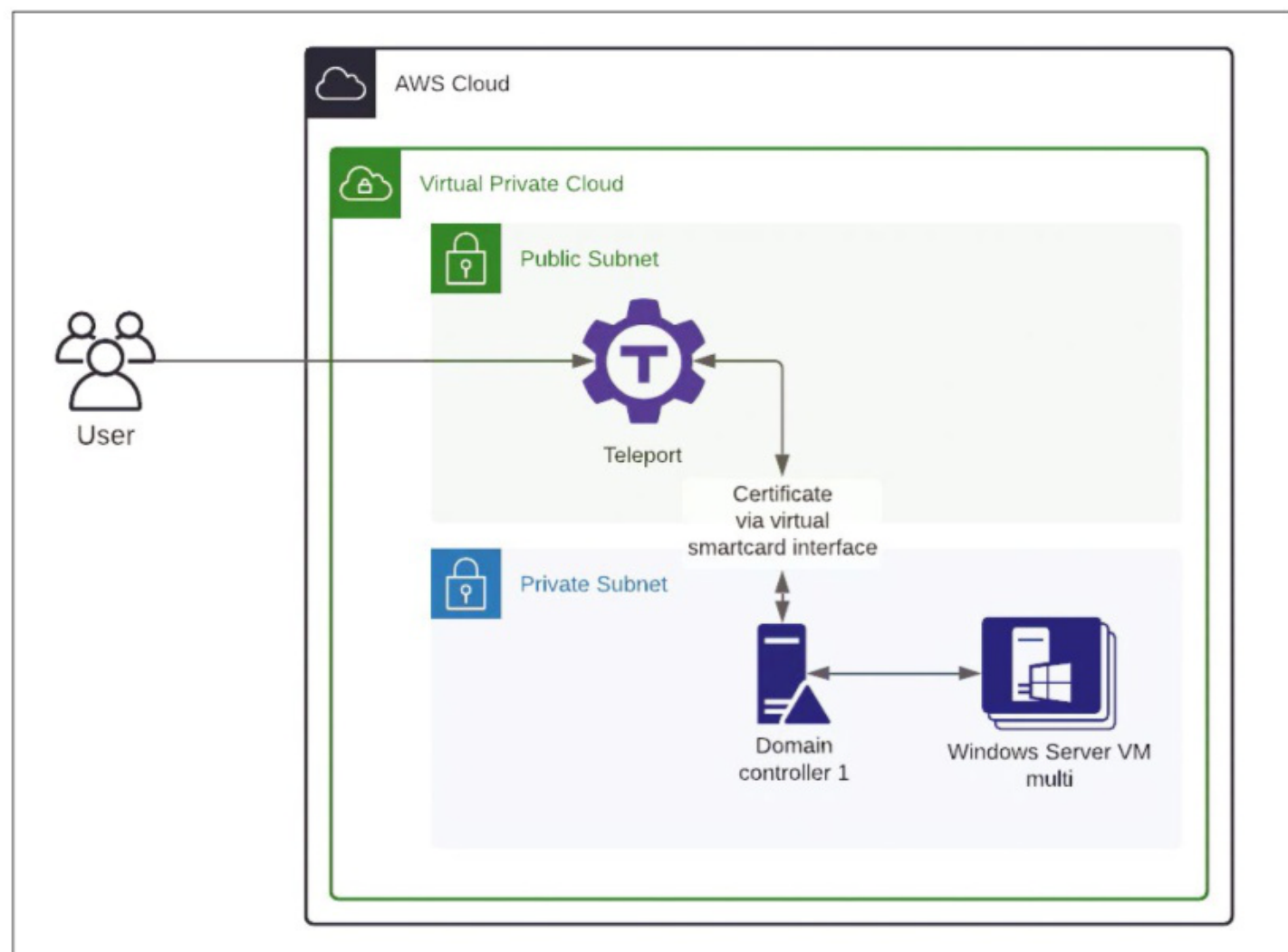


Figure 6: Teleport enables authenticated access to virtual machines in the cloud if combined with Azure Active Directory or Azure Active Directory Domain Services. © Teleport

A camel with a colorful saddle and harness stands in a sandy desert. In the background, a city skyline is visible under a clear sky.

Azure AD with Conditional Access

Is It Real?

Trust is good, but controls are better. As more flexible working models become widespread, the boundaries of the classic perimeter are blurring and softening existing models of trust for adopting cloud software and data storage or running domain controllers or core applications in the cloud. By Florian Frommherz

Terminal devices increasingly reside outside the corporate network and can be reasonably trusted on there to access applications and resources – if they use VPNs, multiple-factor authentication (MFA), and certificates. However, if the application, parts of the infrastructure, or the data itself are not on the internal network, a VPN is not a very elegant approach, to put it mildly.

The catch is that the VPN configuration is installed once only – presumably along with a certificate – on a user's smartphone or PC to provide the return channel to the corporate network. Does anyone actually check whether the certificate matches the device, if it has been sniffed, or even if the device is still in use by the user who originally commissioned it? The same applies to the intended use of VPN tunnels: Does the tunnel follow the intended route or does the process have anomalies?

Defining Trust

If all of your devices, users, apps, and resources operate in the cloud,

anomalies are easier to detect, and checking for the normal state is simpler. A learning process that can reliably certify correct and trusted access or identify risks is thus possible. You need to be able to verify that certain conditions for access are met, depending on what object is to be accessed, before you can allow, say, a mailbox to be opened by an email client on a computer outside the corporate network. The mindset behind this is “zero trust,” and it follows the approach of looking closely at a person's access context and making a decision according to the results of the check: The mailbox is then either opened or it remains locked.

Numerous aspects can be checked in the process: Whether the user is known and has completed MFA at login, whether the device is familiar or even considered “healthy,” whether the user and device can be matched to a region that corresponds to the work location by IP or GPS data, whether the access times and work patterns across multiple applications match the employee's normal

behavior, and whether the application accessing the data gives rise for concern.

If the logging and access controls on the internal network are based on the Kerberos protocol, it is difficult to implement zero trust with legacy techniques. Kerberos and NT LAN Manager (NTLM) do not have any mechanisms to evaluate the access context – just the device account, at best. The normal ticket lifetime is also several hours before any re-evaluation of the context is possible. Modern protocols are different. More can be done with tools like mobile device management (MDM), identity providers, and web protocols that interact with the user; more importantly, it can be done permanently because access tokens in the OAuth 2.0 standard are usually only valid for one hour, after which you need to obtain a new token, which involves an in-depth re-appraisal of the access context. At the end of the day, it's all about minimizing risk and protecting your network against unauthorized access and data leakage while maintaining the productivity of mobile or hybrid workers.

Photo by Jeffrey Dungen on Unsplash

Conditional Access Technology

In the Microsoft cloud, the zero-trust concept is built into many components. The control mechanism found at both the Policy Enforcement Point (PEP) and Policy Decision Point (PDP) in the zero-trust environment is Conditional Access, which is part of the token service that issues modern access tokens for Security Assertion Markup Language (SAML), OpenID Connect (OIDC), and OAuth 2.0 applications integrated with Azure Active Directory (AD) [1]. Anyone who wants to access a resource like an Exchange mailbox that is integrated with Azure AD needs a token from the service, and the token is

cross-checked by Conditional Access. Before issuing access tokens for connected resources, an access context check is performed. Azure AD itself can look at the user's account, group memberships, and role assignments. With the help of the Identity Protection feature in Azure AD, advanced risk assessments are possible for the user account itself or the specific login process. For example, Azure AD blocks access from Tor browsers, users who request access tokens almost simultaneously from two distant locations on the basis of the IP address identifiers, or user accounts whose passwords are believed to be known on the dark web.

In addition to Identity Protection, Azure AD uses other data from

surrounding systems (Figure 1).

The compliance status of the device comes from Intune, which can certify the device as compliant with the enterprise configuration and health requirements. Windows AD can send information about devices with classic domain membership; the devices are at least “known” in this case, even if they have not been credibly checked for a compliant configuration. Up-to-date threat information for endpoints is added by Microsoft Defender for Endpoint. Armed with these checks, administrators can formulate rules that describe access, such as:

- Employees can access their mailboxes with Outlook if they come from at least a known or managed and compliant device.
- Employees who belong to the “No Outlook” group can access their mailboxes from a browser if they come from a known or compliant device or have passed through MFA.

Where possible, modern protocols allow interaction between Conditional Access and the user, which means that Conditional Access is not restricted to simple issue or deny token decisions. If trusting the user's end device is inappropriate, the ruleset allows for alternatives. The user can be prompted to use MFA or dragged by a rule into Microsoft's cloud proxy, which blocks all downloads in the session and prevents data leakage. The employee can then access the data – after carrying out an MFA procedure – while IT ensures that no data is leaked to unknown devices. If some members of the workforce do not have compliant devices or are not allowed to use a company smartphone, enterprise IT simply applies the Conditional Access policy and requires MFA and the use of a Microsoft app populated with an Intune configuration to regulate and restrict data leakage, isolation, and copy-and-paste.

For these cases, zero trust can mean that access does not fail in the classic sense of a user being blocked, but that an acceptable and manageable

The screenshot displays the Microsoft Conditional Access policy configuration interface. The left pane, titled 'New', shows the configuration for a new policy named 'Applications - High Business Impact (fat ...)'. It includes sections for Assignments (Users or workload identities), Cloud apps or actions (4 apps included), Conditions (1 condition selected), Access controls (Grant, 0 controls selected), and Session (0 controls selected). The right pane, titled 'Grant', shows the enforcement settings. Under 'Control access enforcement to block or grant access', 'Grant access' is selected. Several controls are enabled: 'Require multi-factor authentication', 'Require device to be marked as compliant', 'Require Hybrid Azure AD joined device', 'Require approved client app', 'Require app protection policy', 'Require password change', and 'Guest User Consent'. A warning message at the bottom right states: 'Don't lock yourself out! Make sure that your device is compliant.'

Figure 1: A sample Conditional Access setting for a critical app, in which applications with sensitive data can only be used in the context of MFA on a trusted device.

risk is achieved with user session restrictions, after which, employees can then go about their work.

Establishing Manageable Rules

Conditional Access is based on rules (policies) that describe permitted access patterns and their consequences, such as access, denial, or restriction. Because the tool allows a large number of data points to be checked and because you need to reconcile a large number of differing requirements, user groups, and applications, the ruleset can quickly become complex. The sooner you have a clear picture of the work patterns, applications, and equipment you want to allow, and in what combinations, the sharper the ruleset focus can be. If you can standardize the rules, that's even better, because a large number of exceptions and special configurations for small user groups will complicate troubleshooting and debugging.

If you are able to create a three-party team consisting of staff from security, application owners, and people from production or the workplace to decide what patterns are allowed and what needs to be avoided, you will benefit from a large amount of crucial input. Security can say something about the risk and protection needs of the resources you want to protect with a Conditional Access rule. Application owners help to classify the application and analyze the data and information expected to be used in it. People who supervise the workplace, tools, and day-to-day operations can provide important input on productivity, single sign-on (SSO) requirements, and app usage in terms of frequency, endpoints, and locations. The hints given by these three groups result in a single classification for each application in an ideal case, specifying the ruleset or the restrictions on use for the application. This process then develops into an enterprise-wide classification that facilitates the assignment of applications to Conditional Access rules:

Once the band of three and IT have reached an agreement, critical applications and data can always follow a specific access pattern. For example, a trusted, healthy device plus MFA opens all doors, and less critical applications are only accessible from trusted devices or with the cloud proxy plus MFA. On the basis of this rough classification with a few rules on top, scenarios can be created for the everyday applications used by most employees. Even a single Conditional Access rule for the classification covering several apps is conceivable. You need to look at exceptions separately: For example, do you have data and apps that need to be accessible to non-trusted devices? If so, is it legitimate for a company to ask employees

to install familiar Microsoft apps on their – potentially private – smartphones to cover these additional scenarios (Figure 2)? Should B2B guests never be able to access critical applications and data as a matter of principle, or is access legitimate with certain restrictions? Another exception could define, for example, to what admin accounts are allowed to connect.

The art of defining correct classifications lies in taking the middle path with as few rules as possible to maintain clarity, and as many rules as necessary to accommodate all relevant exceptions. Conditional Access is the tool that you use to define and implement policies for zero-trust operations – a clear-cut structure and policy helps you create a strict set of

The screenshot displays the 'New' Conditional Access policy configuration in the Microsoft Azure portal. The breadcrumb trail is 'Home > FrickelsoftNET > Security > Conditional Access >'. The policy name is 'Applications - Low Business Impact (BYO...)' with a green checkmark. The description states: 'Control access based on Conditional Access policy to bring signals together, to make decisions, and enforce organizational policies. Learn more'. The configuration is divided into several sections: 'Assignments' (Users or workload identities: 'All users included and specific users excluded'; Cloud apps or actions: '4 apps included'; Conditions: '2 conditions selected'), 'Access controls' (Grant: '0 controls selected'; Session: '0 controls selected'), and 'Enable policy' (set to 'Report-only' with 'On' and 'Off' options, and a 'Create' button). A right-hand sidebar titled 'Grant' allows controlling access enforcement, with options to 'Block access' or 'Grant access' (selected). Under 'Grant access', several controls are checked: 'Require multi-factor authentication', 'Require approved client app' (with a link to 'See list of approved client apps'), and 'Require app protection policy' (with a link to 'See list of policy protected client apps'). Other unchecked options include 'Require device to be marked as compliant', 'Require Hybrid Azure AD joined device', 'Require password change', and 'Guest User Consent'. At the bottom of the sidebar, there are options for 'For multiple controls': 'Require all the selected controls' (selected) and 'Require one of the selected controls'. A 'Select' button is at the bottom right of the sidebar.

Figure 2: Sample Conditional Access configuration for a non-critical app in bring-your-own-device scenarios.

rules that is only softened in case of relevant exceptions.

Feeding Certificate Authority with Data

For Conditional Access to do its job and ensure flexibility in rule creation, you need to provide the PDP with at least the data that you also want to use in the rules. Herein lies the strength of zero trust: Interfaces that incorporate other data, insights, and information for evaluation establish the necessary trust.

User identities, group and role assignments, and devices from AD will give you a good start, and you will easily see which of these are as hybrid Azure AD-joined devices. If you want to connect smartphones or devices without a domain affiliation, you will need to integrate and manage them with Microsoft Intune. After doing so, you can identify devices running macOS, Android, and iOS and have the compliance status shared between Intune and Conditional Access, allowing Conditional Access to apply a *Require device to be marked as compliant* tag. If trusted work locations exist that can be defined by IP subnets or GPS locations, you can store these locations as a trusted location in Conditional Access, which then allows Conditional Access to apply various rules on the basis of the IP address and, for example, waive the need for MFA or device status checks. To handle temporary exceptions, an IP check is useful, but in terms of the zero-trust concept, IP subnets are too easily manipulated to be checked securely. For secure administration, you can also have Conditional Access enforced for admin workstations; then, Conditional Access not only checks

whether an admin is working on a known and permitted device, but if this is also the specified device. This check is known as a device filter and lets you tag a device as a secure admin workstation or a privileged admin workstation. Admins then need to pass a check (e.g., user has an Exchange Service Administrator admin role, is working on a trusted and compliant device, and the device is tagged as a privileged access workstation, or PAW) before they can, say, access Exchange [2].

The more data you provide to your PDP as input and to the PEP for risk mitigation, the more powerful the toolbox you have for verifying trust or establishing it through restrictions or reviews, as appropriate. To mitigate risk for scenarios in the Microsoft cloud, the most important thing is that you deploy MFA and have a strong credential strategy in place, accompanied by self-service password resets in the event of account theft. If you're deeper into the Microsoft cloud, limited sessions and cloud access broker reviews offer another good tool.

Faster Detection of Breaches of Trust

A new concept, already in use in some Microsoft applications, supports even faster responses to breaches of trust. Services can gain deeper integration with Azure AD and Conditional Access insights. With the help of an event subscription, applications learn that a user has been locked out, is at risk, or is working from a different location. The application can then invalidate the access token immediately after the event arrives and send the user

back to Azure AD to request a new token.

This process reduces the time between two Conditional Access checks during token issuing, during which the token is valid and no zero-trust check can take place. Microsoft calls this continuous access evaluation [3], and the technology is already integrated into Exchange, Teams, and SharePoint and will be offered to other manufacturers as standard in the medium term.

Conclusions

Zero trust starts with a new mindset, grows as you create a database for trust decisions, and thrives on a combination of decision and enforcement points. In the Microsoft universe, Azure AD with Conditional Access occupies this place, and it makes use of other components, such as device management, risk assessment, or user roles. The system has long since outgrown the MFA enforcer stage and implements complex rulesets that not only enable zero trust, but also ensures flexible use of critical data, services, and applications that go beyond Office 365. ■

Info

- [1] Widespread Conditional Access rules: [\[https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/concept-conditional-access-policy-common\]](https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/concept-conditional-access-policy-common)
- [2] Device filters in Conditional Access: [\[https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/concept-condition-filters-for-devices\]](https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/concept-condition-filters-for-devices)
- [3] Continuous access evaluation: [\[https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/concept-continuous-access-evaluation\]](https://docs.microsoft.com/en-us/azure/active-directory/conditional-access/concept-continuous-access-evaluation)

Managing Active Directory sites and subnets

Divide and Conquer

Active Directory domains distributed across multiple physical locations with IP subnetting and network configuration allows for replication and universal user logins. By Thomas Joos

One important area of Active Directory (AD) management is replication of domain controllers (DCs), especially across multiple

sites (**Figure 1**). Separate domains for each of the physical locations is not necessary – multiple domains are more complicated to manage in

most cases than multiple locations for a single AD domain. Active Directory recognizes the physical subdivision and adjusts replication to reflect this. For example, AD replication between sites uses data compression and occurs far less frequently than on a local network. Active Directory uses its own service to manage replication automatically within and between sites. This service, known as the Knowledge Consistency Checker, connects the domain controllers of the various sites and automatically creates a replication topology on the basis of defined schedules and site associations. If more than one DC is available at each site, not all are replicated between sites. Intelligent mechanisms detect grouped DCs and control their replication, as well, so that a slower line between sites is not unnecessarily disrupted by AD replication. Each site has bridgehead servers that pass the information from their own AD site to the bridgehead servers at the other sites. In this way, you also minimize data traffic, because not all DCs transmit data externally.

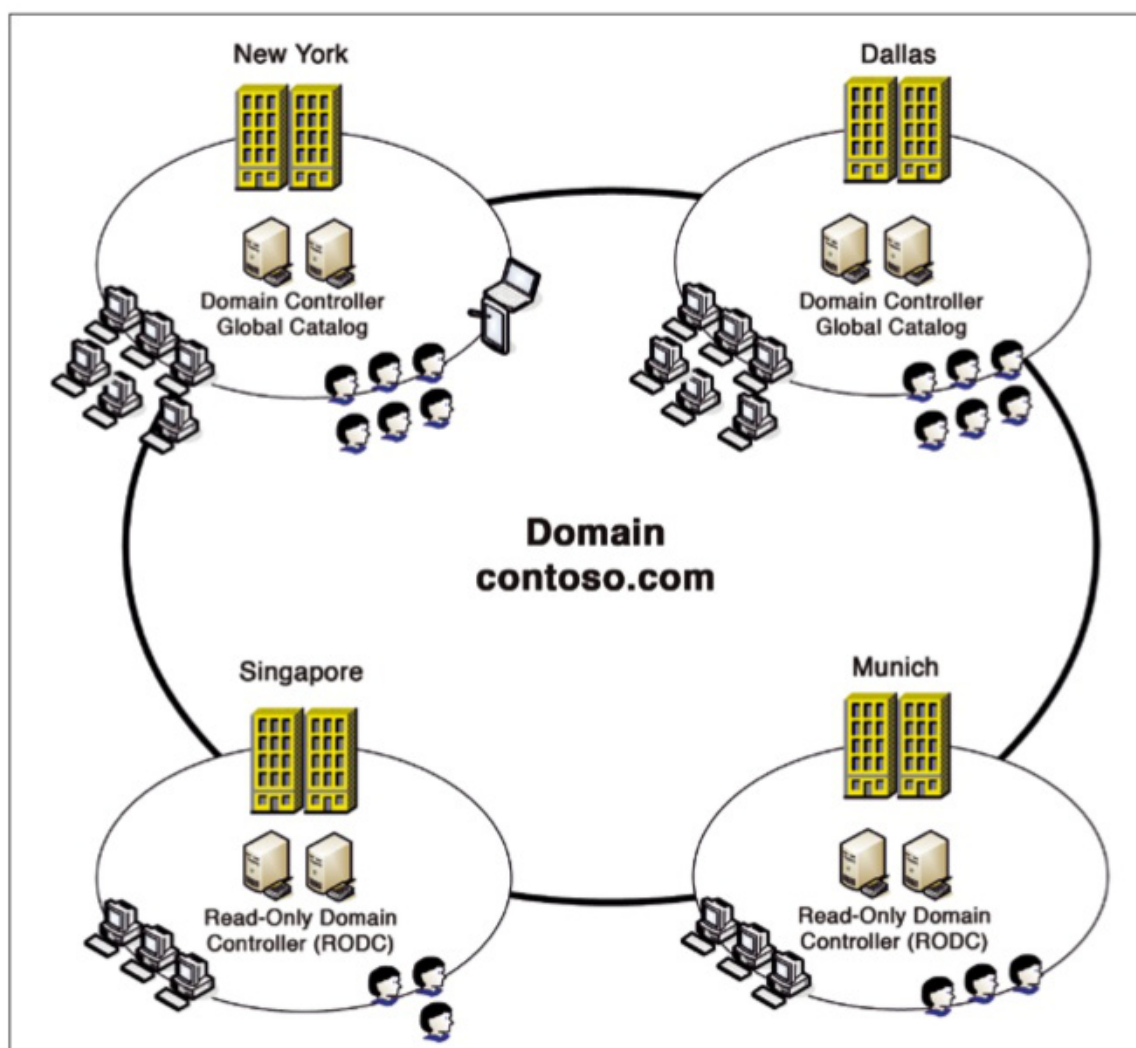


Figure 1: Replication between sites lets you map physically separate networks in Active Directory.

AD Replication Basics

To use replication between sites, you first need to define it and then assign independent IP subnets. These subnets are stored in the Active Directory Management console and distinguish the sites from then on. You can then create links between the subnets and finally distribute the existing DCs to the individual sites. This kind of routing topology means you need to configure the schedules and investigate the cost of the best possible replication.

The management tool you need is the Active Directory Sites and Services snap-in (**Figure 2**). The fastest way to launch it on a domain controller or on a computer with the Remote Server Administration Tools (RSAT) is with `dssite.msc`. Of course, you can also use PowerShell. To create new sites, you must be a member of the Organization Administrators group.

Once you have defined sites and the associated subnets, DCs will automatically be assigned as a function of the subnet to which the IP address belongs. You then need to assign to the right sites existing DCs or servers previously assigned to a location. You can also define these while upgrading the DCs.

To connect each site to the head office, you do not need to use a star topology. Replication in Active Directory also lets you integrate sites that are connected to other sites but not to the head office. Active Directory can manage this, provided you defined the sites and subnets correctly.

Creating New Sites

When you open the Active Directory Sites and Services snap-in, below the Sites entry you will see *Default-First-Site-Name* as the first site. Active Directory always automatically has one site. In the first step, make sure you assign a meaningful name to this site; you can do so in the context menu.

Next, create the additional sites where you want to install DCs by

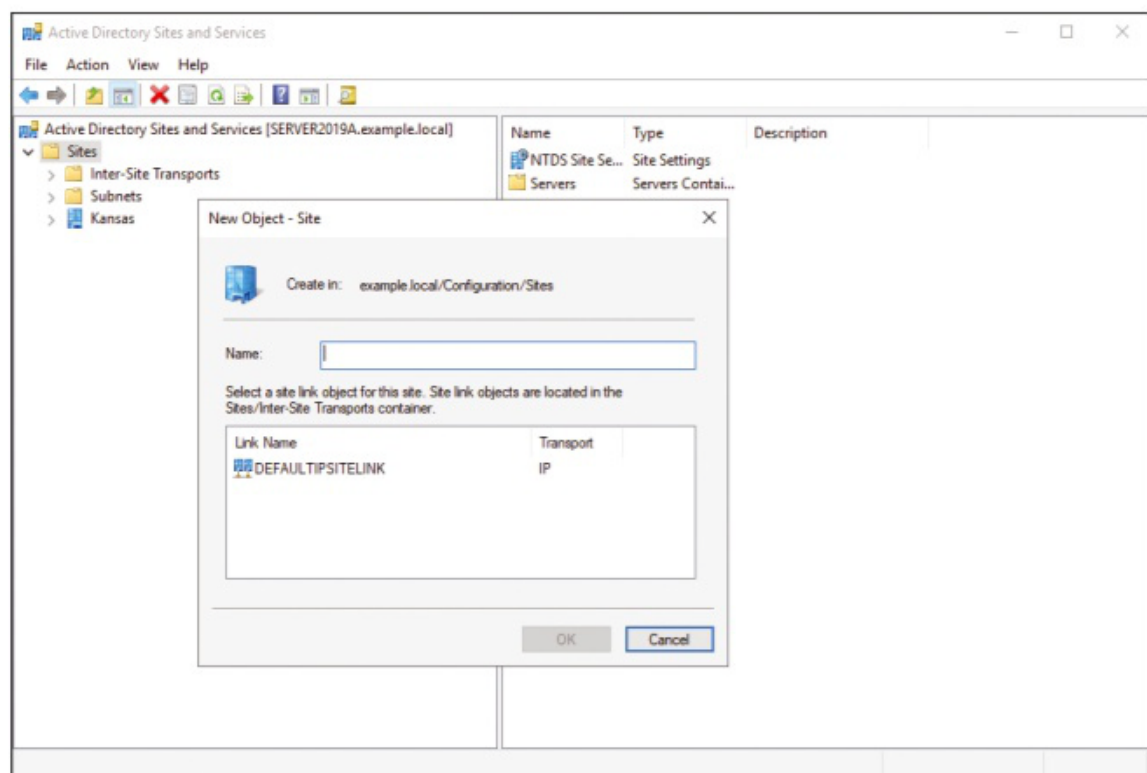


Figure 2: Creating new AD sites in the Active Directory Sites and Services snap-in.

right-clicking *Sites* in the snap-in and selecting *New Site* from the context menu. In PowerShell, use the

```
New-ADReplicationSite <site>
```

command. At this point, no extensive configuration is required for the time being, and if you don't have a DC at the site, creating sites will not affect replication in Active Directory. When you create a new site in the snap-in, a window opens in which you need to specify its name and the site link. You can use the link to manage replication. The settings for this can be changed at any time. By default the *DEFAULTIPSITELINK* link already exists. Now create a list of locations with a CSV file that starts with the line name. It can then be imported with the command:

```
Import-Csv -Path C:\newsites.csv | ?  
Import New-ADReplicationSite
```

Creating IP Subnets

After creating the sites, create the IP subnets and assign them to the respective sites. To create a new subnet, right-click *Subnets* in the snap-in and select *New Subnet* from the context menu.

If desired, you can create IPv6 subnets. After you create the subnet, the console displays it below the menu

item of the same name. At this point, you will also want to create IP subnets on which no domain controllers are installed but that member computers that log on to the DC may be running. You need to assign them to the appropriate sites.

If you click *Subnets*, you will see all the IP subnets and their assigned sites on the right. You can assign subnets to specific sites at any time in the subnets' properties. Sites can be created retroactively, and new subnets can be assigned to existing sites.

Creating Site Links

After you have created sites and the IP subnets there, it's time to set up the site links. If your lines have different bandwidths, it may make sense to create different site links. You can then schedule when replication is possible as a function of the site links and create site associations on the basis of IP or SMTP, although IP is used in most cases.

To establish a new site link, under *Inter-Site Transports* right-click on *IP* | *New Site Link*. When assigning the name, it is best to use a designation that allows conclusions to be drawn about the respective locations (e.g., *Liverpool*, *Manchester*) or the type of connection between the various branches. At this point, you can also decide which sites to connect with

this site link. A site can be a member of several links.

Replication takes place through those site links with the lowest defined costs. If you click on *IP*, you will see the site links on the right. Once you have created the site link, its properties can be edited. On the *General* tab, define the interval at which you want to replicate the information between the sites. By default, replication is set to three hours and the costs are set to 100. If you click the *Change Schedule* button, you can specify the times at which replication is allowed with this site link.

At this point you can also create site link bridges, which are used by Active Directory if two sites have no physical connection but are connected by a third site through which you want AD replication to take place. These bridges are created automatically. If you want to do this manually, you have to disable the automatic mechanism under *Inter-Site Transports* by right-clicking *IP* | *Properties* and checking the box beside *Bridge all site links*.

Site links can also be set up in PowerShell,

```
New-ADReplicationSiteLink 2
CORPORATE-BRANCH1 2
-SitesIncluded CORPORATE,BRANCH1 2
-OtherAttributes @{options='1'}
```

and the cost and time frame for synchronization is defined with the command:

```
Set-ADReplicationSiteLink 2
CORPORATE-BRANCH1 2
-Cost 100 2
-ReplicationFrequencyInMinutes 15
```

Assigning Domain Controllers

Any DCs that are already installed to the correct site need to be moved manually. To do so, right-click on the server in the Active Directory Sites and Services snap-in and select *Move* in the context menu. DCs can also be moved to new locations in PowerShell,

```
Get-ADDomainController <name of server> | 2
Move-ADDirectoryServer 2
-Site <name of site>
```

or you can drag and drop them to different locations. Windows sets up the replication links automatically. To see these, select *Sites* | *< Site >* | *Servers* | *< Server >* | *< Server-name >* | *NTDS Settings*. You can set up manual connections here by selecting *New Active Directory Connection* from the context menu. In PowerShell, you can display the replication connections, display detailed information about the individual sites, display only the name, and get a list of DCs and sites:

```
Get-ADReplicationConnection
Get-ADReplicationSite -Filter *
Get-ADReplicationSite -Filter * | ft Name
Get-ADDomainController -Filter * | 2
ft Hostname,Site
```

If replication problems occur in Active Directory, first make sure the DCs experiencing replication difficulties are configured for the correct site. To do this, type the command

```
nltest /dsgetsite
```

at the command prompt.

Knowledge Consistency Checker

Once you have created the routing topology, the Knowledge Consistency Checker (KCC) automatically sets up the links between the DCs. KCC automatically configures AD replication according to the sites, the links, the schedules and costs, and the DCs that exist there. The service is completely automatic and runs on each domain controller in the forest. It does not link every single DC to every single other DC; rather, it sets up an intelligent topology.

KCC checks every 15 minutes that the existing connections are working and automatically changes the replication topology, if needed. Within a site, the service creates a ring topology, attempting to have no more than three other DCs between two separate domain controllers.

During data transfer between various sites, the AD data is not transferred by all DCs to DCs on other sites (as previously mentioned) but only by one DC at a time. This DC, known as the bridgehead server, automatically replicates with other bridgehead servers on other sites (Figure 3). At the respective site, the domain controllers in turn replicate with each other. KCC automatically determines which DCs at a site become the bridgehead servers. The selection of the bridgehead servers at a site is handled by the intersite topology generator (ISTG), which is part of KCC (Figure 4).

Configuring ISTG

If you click on a site in the snap-in, the entry *NTDS Site Settings* appears on the right side. If you call up the properties of this entry, you will see

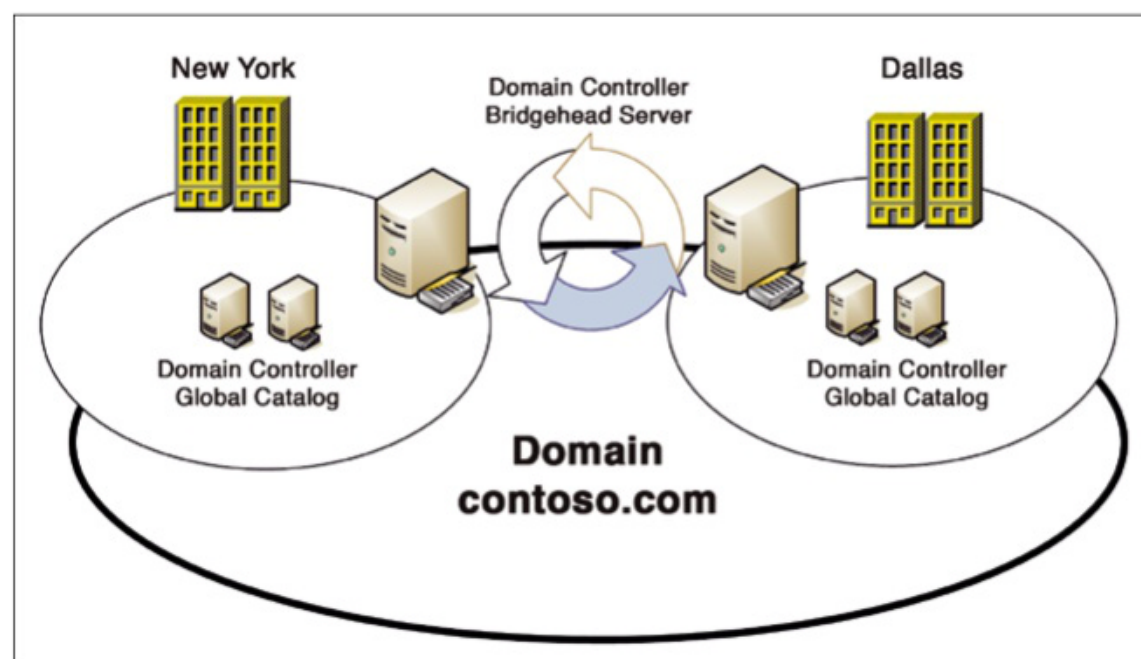


Figure 3: Bridgehead servers handle replication between sites.

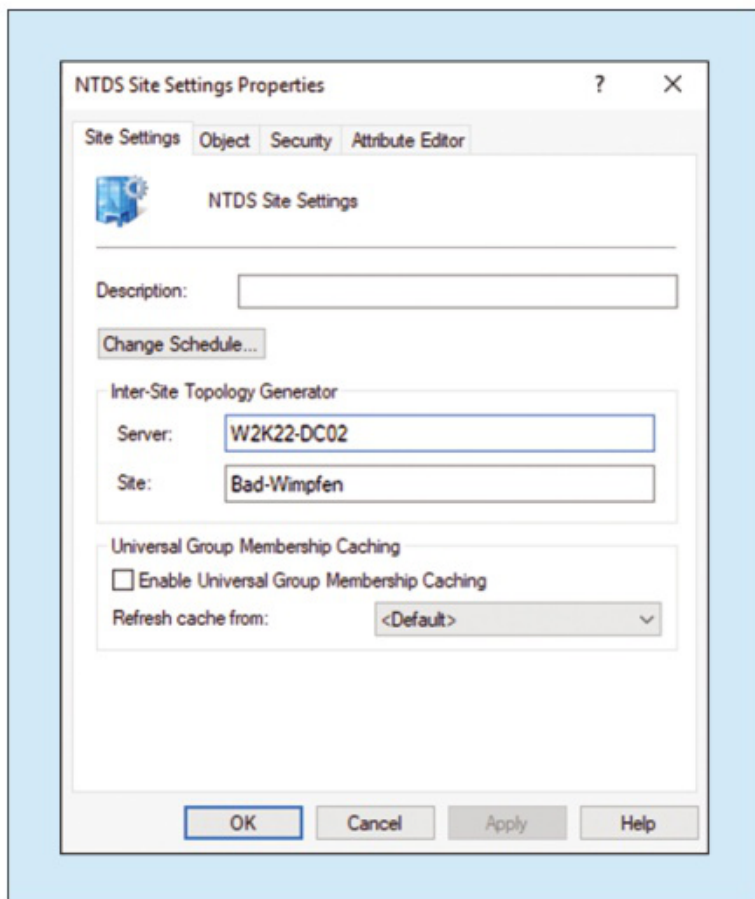


Figure 4: You can display the ISTG of a site in the NTDS Site Settings Properties dialog.

the current ISTG in the Inter-Site Topology Generator section.

At this point you can also customize the *Enable Universal Group Membership Caching* option. The group membership of these groups is part of the global catalog in Active Directory. If you do not operate a global catalog at the site, other DCs can be configured to store these memberships with this option. If you made changes to the routing topology, you have the option of enabling them immediately. To do this, proceed as follows:

- Open the Active Directory Sites and Services snap-in.
- Navigate to the site from which you want to start the scan.
- Click the current ISTG role owner of the site.
- Right-click *NTDS Settings* and select *All Tasks | Check Replication Topology* from the context menu.

If you right-click and select a line, you can trigger replication to this server immediately with the *Replicate Now* option, but if you choose to replicate to a DC that resides at a different site, replication will not start immediately. Instead, replication waits until the next time allowed by the schedule. Before replicating data, the DC first makes sure it can connect to the

other DCs. The server indicates the successful connection. If the replication partner is unreachable, an error message is displayed.

Fixing Replication Errors

If you have a problem with AD replication, always perform full diagnostics on the DCs. It is best to make a simple sketch of the replication connections of the DCs and record exactly which ones can no longer replicate with which other DCs. A sketch like this will usually help you identify quickly which

DC is the root cause of the problem. Then, you can take a closer look at it and check whether it is working within its site. The next step will be to look at the Event Viewer and the Directory Services event log. In particular, look for errors from NTDS KCC, NTDS Replication, or NTDS General. PowerShell is also a convenient way to troubleshoot. The familiar *repadmin*, *dcdiag*, and *nlttest* tools, as well as others, are available in Active Directory for this purpose. Microsoft shows you exactly what troubleshooting can look like in an example online [1]. Before you use tools to examine replication in more detail, however, you will first want to rule out the most serious and most frequent causes of error:

- Can a general problem on the DC that can no longer replicated be discovered with *<dcdiag>*? If so, maybe the problem is not related to replication, but the DC is simply malfunctioning.
- Has any software been installed on the DC that could be interfering with replication, such as security software, a virus scanner, a firewall, or something else?
- Does a hardware failure on the DC prevent replication?

- Does a line, router, or firewall have a problem?
- Can the corresponding DC still be pinged, and does the DNS name of the server resolve?
- Are general problems with authentication between DCs logged with *access denied* messages?
- Are the replication intervals between sites set to so short an interval that the previous replication has not been completed by the time the next round starts?
- Have any changes been made to the routing topology that could prevent replication?

The most important tool for checking replication in Active Directory is *repadmin*. To display all the AD replication operations that have occurred, along with errors that might show you what is causing replication to fail, enter:

```
repadmin /showreps
```

You can also display only the errors, redirect the display to a file, and send the replication information to a CSV file:

```
repadmin /showrepl /errorsonly
repadmin /showreps >c:\repl.txt
repadmin /showreps * /csv > reps.csv
```

If a replication connection does not work, you need to read the server globally unique identifier (GUID) for each server with the

```
repadmin /showreps
```

command. Each server shows the directory service account (DSA) object GUID in the window. You need to reference this to add a connection, then use the GUID in the *repadmin /add* command line. The domain name for the example here is *contoso.int*. The server GUIDs for the two DCs are:

DC1 GUID = e8b4bce7-13d4-46bb-b521-8a8ccfe4ac06

DC5 GUID = d48b4bce7-13d4-444bb-b521-7a8ccfe4ac06

In the Active Directory Sites and Services snap-in, delete all connection objects, then create a new connection from the broken DC to a working DC with the command:

```
repadmin /add *
„cn=configuration,dc=contoso,dc=int“ *
e8b4bce7-13d4-46bb-b521-8a8ccfe4ac06.*
_msdcs.contoso.int d48b4bce7-13d4-444bb-b521-7a8ccfe4ac06._msdcs.contoso.int
```

In your environment, of course, you need to use your own server GUIDs and domain name. The rest of the input is identical. During this action, you see an *8441 (distinguished name already exists)* error. Now trigger full replication through the connection you created:

```
repadmin /sync cn=configuration,*
dc=contoso,dc=int DC1 e8b4bce7-13d4-46bb-b521-8a8ccfe4ac06 /force /full
```

After doing so, go to the snap-in and make sure you again have automatically generated connection objects from the failed machine to the working DC. The

```
repadmin /replsummary /bydest
```

command is useful for displaying errors by replication target. If you want to show the errors by the replication sources for these the targets, use the command:

```
repadmin /replsummary /bysrc
```

to get a picture of which inbound and outbound replication actions are not working properly. To check whether replications are suspended on a DC because communication with other DCs is impossible, use

```
repadmin /queue
```

If replication is working, the queue should be processed and approach zero.

Testing Replication with PowerShell

You can also discover the replication status in PowerShell:

```
Get-ADReplicationUpToDatenessVectorTable * | Sort Partner,Server | ft Partner,Server,UsnFilter
```

The asterisk in place of <name of server> displays a list of all servers. To view the individual sites and their domain controllers, enter:

```
Get-ADReplicationSite -Filter * | ft <name>
Get-ADDomainController -Filter * | ft <hostname,site>
```

The `Get-ADReplicationFailure` cmdlet lets you check for replication failures in PowerShell. It expects the `-Target` parameter and the name of the domain controller to be checked:

```
Get-ADReplicationFailure -Target dc01
```

If you want to check multiple domain controllers, add the names in a comma-separated list. The command

```
Get-ADReplicationFailure *
-Target „joos.int“ -Scope Forest
```

defines a complete AD forest as the target.

Conclusions

Subnet management plays an important role in Active Directory. Administration is ultimately easier if a domain spans multiple sites and you don't have to create a separate domain for each site. Configuring subnets is not a complicated matter, but it does require some attention to detail. By the way, Windows Server 2022 manages these operations in the same way as in previous versions. ■

Info

[1] Troubleshooting AD replication error 8589: [\[https://docs.microsoft.com/en-us/troubleshoot/windows-server/identity/replication-error-8589\]](https://docs.microsoft.com/en-us/troubleshoot/windows-server/identity/replication-error-8589)

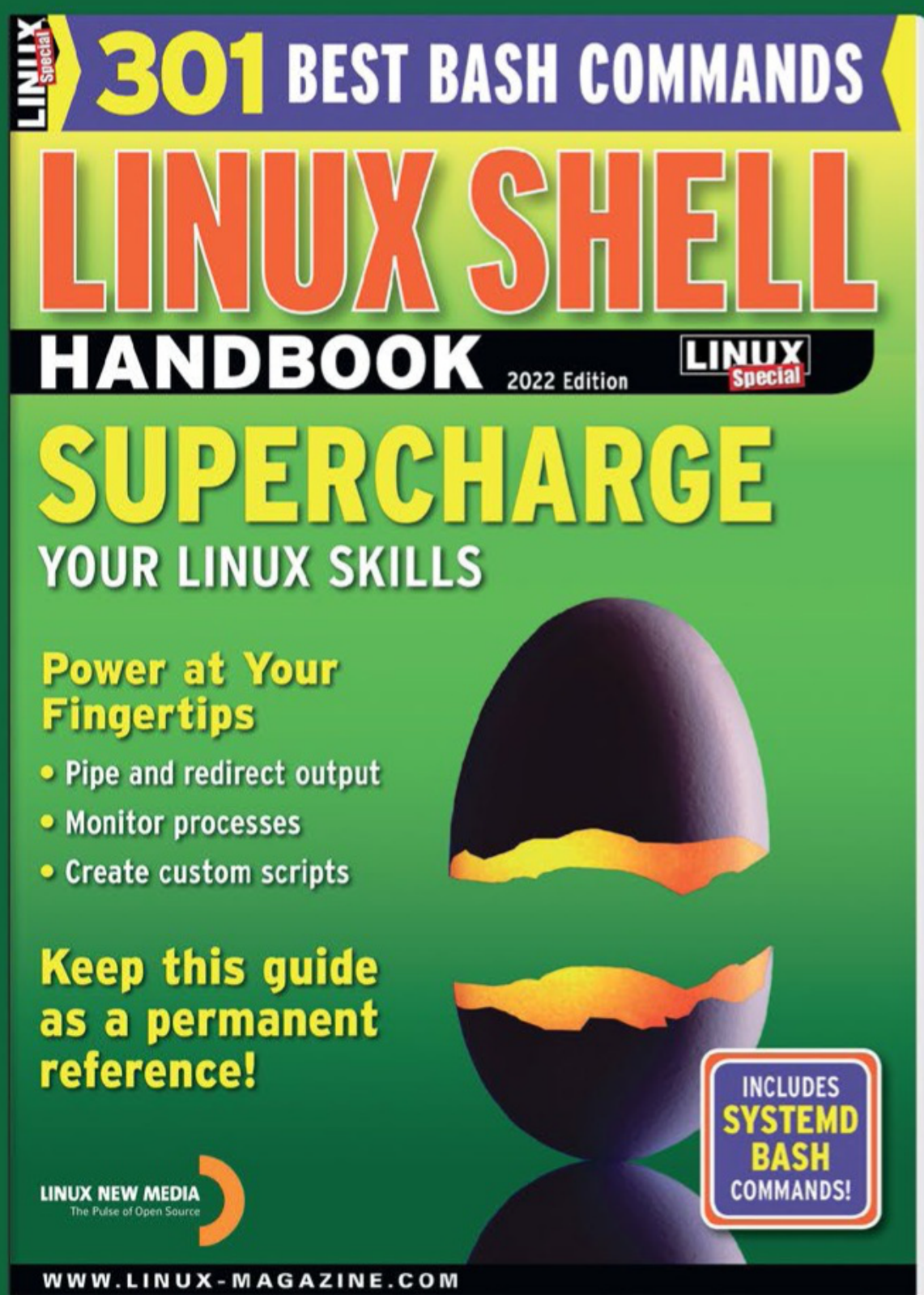
The Author

Thomas Joos is a freelance IT consultant and has been working in IT for more than 20 years. In addition, he writes hands-on books and papers on Windows and other Microsoft topics. Online, you can meet him on [\[http://thomasjoos.spaces.live.com\]](http://thomasjoos.spaces.live.com).

THINK LIKE THE EXPERTS

Linux Shell Handbook 2022 Edition

This new edition is packed with the most important utilities for configuring and troubleshooting systems.



Here's a look at some of what you'll find inside:

- Customizing Bash
- Regular Expressions
- Systemd
- Bash Scripting
- Networking Tools
- And much more!

ORDER ONLINE:

shop.linuxnewmedia.com/specials



Automate macOS 12 with the Shortcuts app

Little Helpers

Apple ported the Shortcuts automation tool known from the iPhone and iPad to macOS Monterey 12 to help users make their everyday work more convenient, with or without programming knowledge. By Christian Knemann

Low-code or, even better, no-code solutions are en vogue. These products are designed to enable users to automate small tasks without programming or scripts, removing the need for tedious manual work. Apple Shortcuts are already old acquaintances on iOS and iPadOS. Whereas users formerly needed to download the Shortcuts app from the App Store manually, it became an integral part of Apple's mobile operating systems in version 14. With macOS Monterey 12, Apple has now also added Shortcuts to its desktop operating system.

Almost Identical

Apple helps newcomers with a brief introduction [1]. If you have already familiarized yourself with the app on mobile devices, you will immediately feel comfortable with it on macOS, too. Apps use Apple's SwiftUI GUI framework on both platforms and share the codebase. Because you can store your individual shortcuts in iCloud, many of your automation shortcuts can be used across both platforms – as long as you use actions that exist for both macOS and

iOS. If the feature set is not enough, you can use AppleScript, JavaScript for Automation (JXA), or shell scripts to extend the shortcuts on macOS – but more about that later.

Importing Automator Workflows

Shortcuts is not Apple's first attempt to automate macOS. The Automator app has been an integral part of the operating system since Mac OS X 10.4 (Tiger), which was released in 2005. In recent years, however, Apple has not seriously maintained this application. In direct comparison to Shortcuts, the user interface seems pretty much outdated, and the learning curve is steeper. For the time being, the two apps coexist, but Apple is not expected to continue support permanently for both automation tools in parallel. If you have already created workflows in Automator, you can export them to a *.workflow type file and try to import them into the Shortcuts app [2]. Unfortunately, this method does not always work. In our lab, Shortcuts did not want to adopt a simple workflow that renamed and enumerated files

and, for want of a meaningful error message, did not reveal what caused the failure.

If you are new to macOS automation, you can boldly go straight to Shortcuts, because it handles many of the actions specific to the operating system and helps integrate scripts from Automator.

Sharing over iCloud

When you launch Shortcuts from the Launchpad, the app comes up with a two-part layout featuring a navigation bar to the left and a main area that displays the available shortcuts as tiles. Before I turn to working with Shortcuts, I'll first take a look at the global options in the *Shortcuts | Preferences* menu. In the General section, iCloud synchronization is enabled by default, which ensures that you can use your shortcuts on all your devices. iCloud has the option of sharing your efforts with the world. The Shortcuts app generates an iCloud link for each shortcut, and you can share this in any way you want (e.g., by AirDrop, email, or text message). Beware, though, this link is public. Anyone

who discovers it can download and use your shortcut in the Safari browser.

Alternatively, you can use *File | Export* to save a shortcut as a file with the `.shortcut` extension, which you can then choose to pass on to the whole world or only to people you know [3], in which case, the contact information of the respective recipients must be in your Contacts on macOS. Importing a shortcut like this takes you back to the *General* tab in *Preferences* because you need to enable the *Private Sharing* option there [4].

Organizing Shortcuts in Folders

The *Sidebar* pane of *Preferences* controls which subfolders you see under *My Shortcuts*. In addition to dynamic folders for recently changed or executed commands, the app provides folders for commands that work on the Apple Watch and commands for the share sheet (i.e., the share icon, but only on iOS and iPadOS). Although the Finder and Safari on macOS also offer a share icon, shortcuts from this category do not appear there. The *Quick Actions* options for the Finder and Services menu and the *Menu Bar* are particularly interesting on macOS. Alternatively, you can launch shortcuts from a Spotlight search or Siri.

If the basic actions of the shortcuts are not enough for you, the *Advanced* tab in *Preferences* will be useful. You can enable the *Allow Running Scripts* option to extend your shortcuts with AppleScript, JavaScript, and shell scripts. You can also let shortcuts delete without confirmation, as well as share and delete large volumes of data. However, Apple's documentation does not reveal the exact point at which a data volume is deemed to be large.

In the browser, you can structure your shortcuts by right-clicking in the free area to create individual files in *Folders*. Finally, I'll look at one difference between the mobile operating systems and macOS: On iOS and iPadOS you find the *Automation*

icon, which lets you execute short commands locally on the iPhone and iPad, as well as with Apple HomeKit as a function of states or events, such as at a certain time of day, when entering or leaving a certain place, or even as the result of a system-related event (i.e., wireless or Bluetooth connections, the state of charge or connection to the mains, app-specific events, or changes to focus settings). Unfortunately, this whole concept is completely missing on macOS, but the desktop operating system does offer numerous additional actions that its mobile counterparts lack.

Loading Commands from the Gallery

Regardless of the platform, you will find the *Gallery* section, where you can download to your personal library ready-made shortcuts that are divided into different categories and either use them as is or customize them up front to suit your needs. Blogs and forums on the Internet also provide more or less useful examples, but you will want to check them carefully before you execute them.

For now, the Gallery is a good place to start. Download some examples to get familiar with their functions and structure, and with working with the editor. Once you have loaded a shortcut, you will find it in your collection. If you hover over it, you can run it immediately from the small play icon at the top right of the tile. A double-click on the tile opens the respective shortcut in the editor.

To begin, try two simple commands that work on both iPadOS and macOS. The *Split Screen 2 Apps* shortcut has just one action that does exactly what the name promises. The action expects two variables for the apps to be started. You can statically define two apps when downloading, or alternatively, each time you start the program, you can select which two apps you want the command to start. You can also change your choice in the editor later.

The *Search GIPHY & Share* shortcut already has two linked actions. The

first action opens a free text field for entering a search term and then shows the results in the *Share* action, where you can share one of the matches with another app.

Initial Orientation in the Editor

The editor has the navigation bar on the right side; it is divided into the Action library and the Shortcut Details. You can switch between the two areas with the icons right at the top. The action library is again divided into two registers: You can see the basic actions in the *Categories* tab sorted by their respective application purpose, as well as all the applications that offer actions in the *Apps* tab. Most of the apps are by Apple, with hardly any third-party providers. Apple does offer a suitable interface in the form of SiriKit, which combines the Intents and IntentsUI frameworks. However, developers would actively have to use them and incorporate them into their applications. One hopes more third-party providers will discover Shortcuts for themselves in the future. Until then, advanced techniques will help you target apps outside of Apple's ecosystem. The existing actions prove to be practical.

Anchoring Commands in the OS

Before I show you how to build more complex shortcuts, you should take a look under the Shortcut Details icon (three slider bars) at top right (Figure 1). The *Details* tab lets you choose where each shortcut will be anchored. The options at the top are relevant for macOS. *Keep in Menu Bar* makes the command available in the operating system menubar at the top of the screen. *Use as Quick Action* anchors the command in the *Finder*, which lets you access the command from the *Quick Actions* menu item in the context menu of folders and files, the *Services Menu* of an app, or both. If you have a MacBook Pro with a touchbar, you will have an option for that, too.



Figure 1: The tabs under the Shortcut Details icon lets you customize your shortcuts.

If commands need access to resources locally or on the network, they request them on first use. You will find any permissions granted in the *Privacy* tab, where you can revoke individual permissions or reset data security completely. In the last tab, *Setup*, you can prepare your own creations by means of prompts for sharing and importing by other users. The *Split Screen 2 Apps* shortcut makes use of this. The editor offers even better options when you design your own shortcuts.

Zip Files and Folders

The first short example reads folders or files from the Finder and bundles them into a ZIP archive, which it drops in a target folder. If you click the plus icon in the toolbar, the editor comes up with an empty workspace. In the window

specify what the command should do if no matching input is found.

Now look for the *Make Archive* action in the Documents category of the Action library and drag it into the workspace. The action automatically connects to the previous one, which means it accepts the input for processing downstream. Besides ZIP, other archive formats are available for selection. Next, you need to tell the command what to do with the archive. To do this, drag the *Move File* action into the workspace. It uses the Shortcuts folder as the target by default. Click the appropriate tag, select *Replace* from the context menu,

header, choose a descriptive name and a suitable icon. Now enable the *Use as Quick Action* option under the shortcut *Details* tab and optionally enable *Finder* integration (Figure 2). The first action then appears in the workspace; it initially accepts arbitrary input. You can click on the *Any* tag to limit the accepted input to media and documents and

and set a folder of your choice as the target.

Now when you right-click a folder or file in the Finder and choose *Quick Actions* from the context menu, if your new shortcut does not show up there, you can activate it in the *Customize* option. The command will then appear in the quick actions and bundle the transferred files or folders into the ZIP archive, which it delivers to the desired target.

If you want to prepare the command for sharing with other users, add an import question in the *Setup* section of the Shortcut Details. The main editor window changes and you can select which element the command will prompt for at import time. Now determine the *Folder* for the *Move File* action. If you share the command over iCloud or as a shortcut file and another user imports it, they will be prompted to select an individual target folder.

Connecting and Disconnecting Shares

Everything works, in principle, even if the target is a Server Message Block (SMB) share, but to demonstrate the possibilities of shortcuts, I'll duplicate the command and rebuild it so that it explicitly connects to a server by SMB at the beginning, then writes the ZIP archive to a share on the server and disconnects from the server again.

To do this, drag the *Connect to Servers* action to the second slot on the

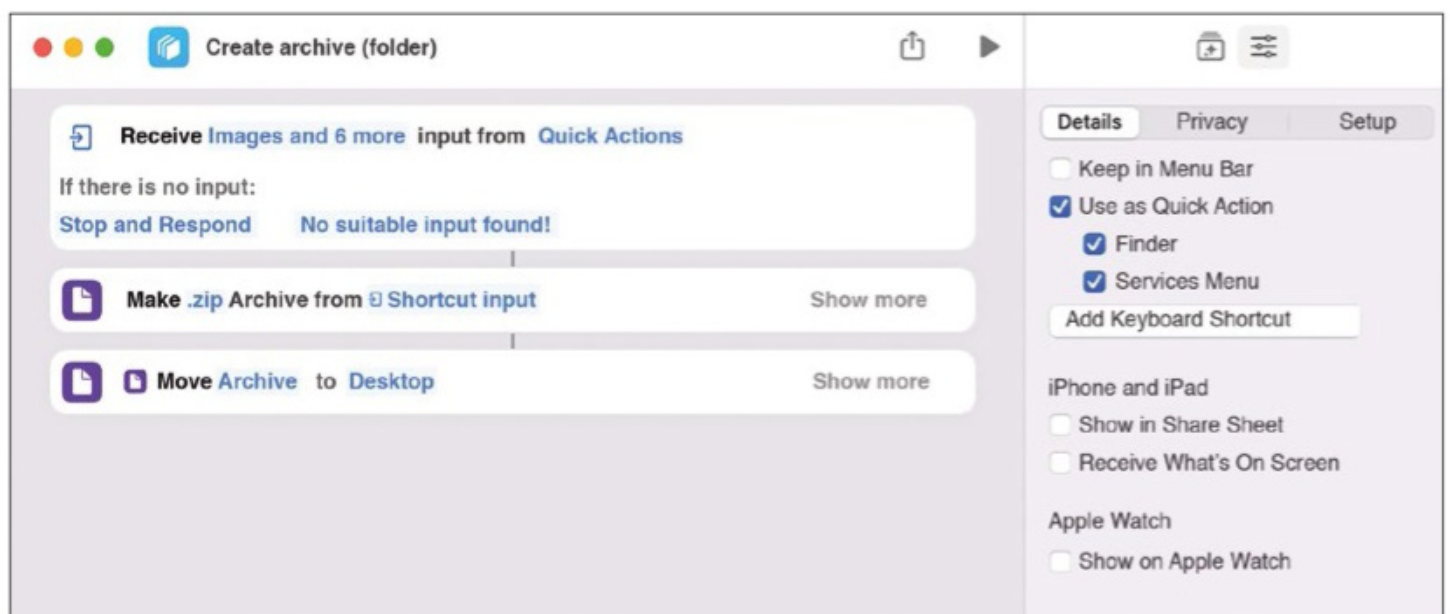


Figure 2: Shortcuts anchor themselves in the operating system menubar or as a Quick Action.

desktop and enter the path to your share as the target with the `smb://<server>/<share>` format. You now want the last action in the command sequence, *Move File*, to adopt the target dynamically. To do so, right-click on the target folder's name in the last block and select *Magic Variable* from the context menu. The display in the editor will then change and you can click on the *Connected to Servers* item farther up in the schedule.

To leverage this action a second time, drag the *Eject Disk* action into the workspace. Again, use a magic variable here that dynamically reads the output of a previous action and therefore automatically the *Connected to Server* return value. In the Finder you can see for yourself how the shortcut works. It should now contact your server, write the ZIP archive to it, and then break the connection. With the help of magic variables you can flexibly access the return values of previous actions.

Sending Email with Apple Mail

For the next example, I'll duplicate the first short command again and modify it so that it doesn't write a ZIP archive to the filesystem, but emails a ZIP instead. To do this, delete the last action and drag the *Send Email* action from the Sharing category to the desktop instead. The action automatically accepts the archive from the previous action as its input, but it has two additional parameters. One parameter defines the subject line for the

new email and the other the recipient. However, the shortcuts being so closely linked to Apple's ecosystem turns out to be a disadvantage. You can only specify a receiver that you previously added as a contact in the macOS Contacts app. If you want to use an email address that does not exist in your contacts as free text instead, you can do so with a simple trick: Drag an action of type *Text* to the second-to-last position, in which you enter the target address. In the last action, you can then turn the recipient field into a magic variable and parse the text from the previous action.

If you now apply the action to files or folders in the Finder, an email with the ZIP archive as an attachment should automatically open. With the help of the text and magic variables, you have freed yourself from the dependency on Apple's Contacts app; however, Shortcuts inherently only looks to use Apple's native mail app for sending.

Scripts

If you prefer to use Microsoft Outlook, you can overcome the barriers, too, by just duplicating the original

Listing 1: toggle_powermode.sh

```
POWERMODE=$(pmset -g | grep lowpowermode | tr -dc ,0-9')
if [ $POWERMODE == 0 ]
then
    sudo pmset -a lowpowermode 1
else
    sudo pmset -a lowpowermode 0
fi
```

shortcut again and deleting the last action. In its place, drag in the *Run AppleScript* action from the Scripting category (Figure 3). If you now apply this command to files or folders in the Finder, a suitably parameterized email with a ZIP attachment is automatically opened in Outlook. Shortcuts can also integrate shell scripts in a similar way. For example, create another shortcut and pin it to the menubar. You want this command to turn off the operating system's power-saving mode if it is on, and vice versa. To do this, drag the *Run Shell Script* action into the workspace. Configure Bash as the shell and enable the *Run As Administrator* action. In the text box, enter the code from Listing 1.

You can then access your new command from the shortcut icon in the operating system's menubar and

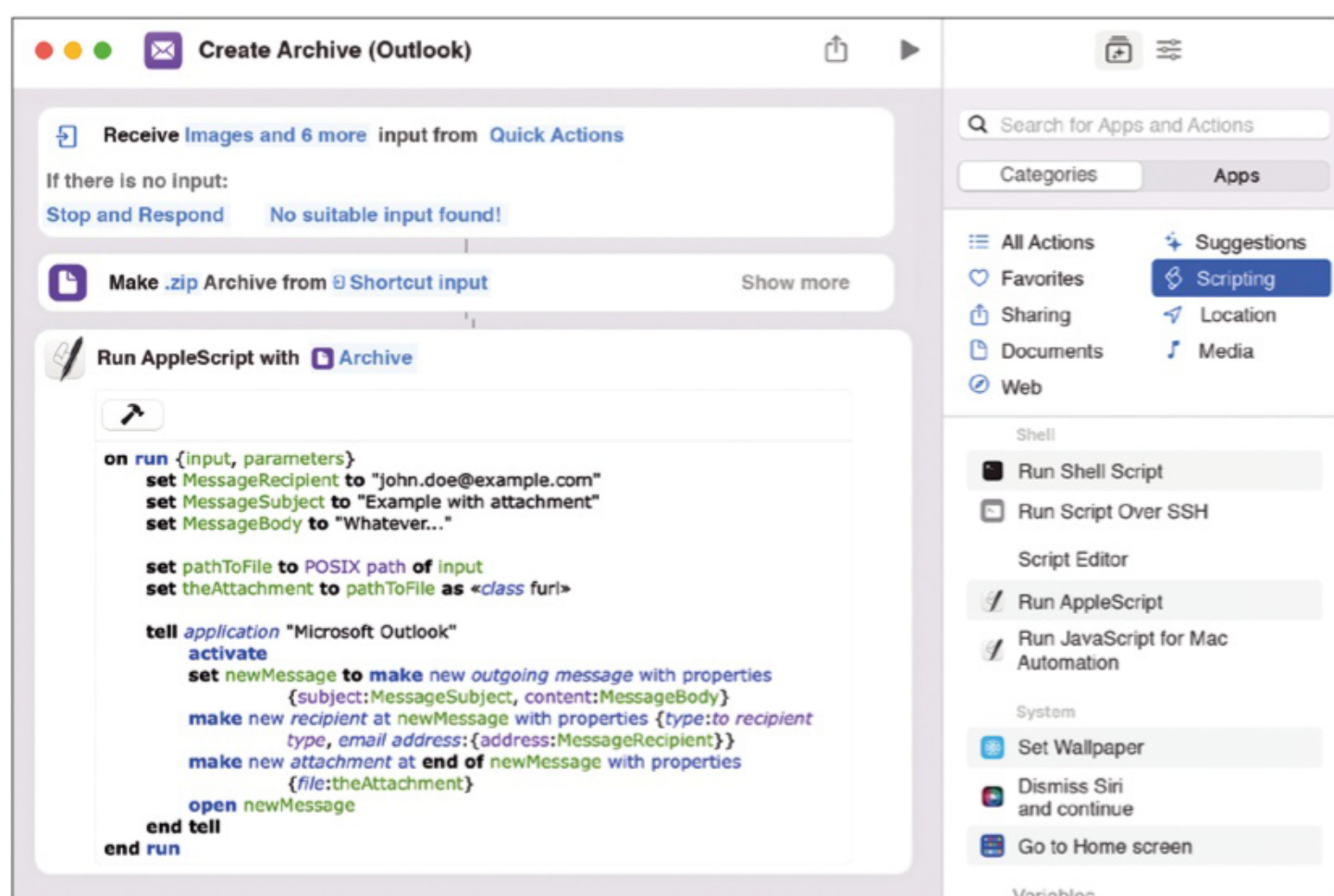


Figure 3: AppleScript opens a new email in Outlook instead of Apple Mail on demand.

toggle your device's power-saving mode with just two clicks.

Executing Loops

The next two shortcuts receive images, scale them, and then write the modified images to a target folder. To do this, again create a new command to use as a quick action and restrict the input to images. Drag the *Get Images from Input* action to the workspace (Figure 4). For the new construct, you need a loop that processes each image that is passed in. From the Scripting section, select *Repeat with Each*. A loop block appears that automatically accepts the retrieved images. Drag the *Resize Image* action into the loop block. You can choose between a fixed value for

the width, height, or longest edge; alternatively, you could use a percentage. As a further building block you can add the *Move File* action to the loop to write the respective image to the desired target folder. Next, pass in some images to the command in the Finder to see whether it works. Shortcuts also supports a simple *Repeat* loop construct with a selectable number of runs, *If* blocks, and a simple *If – Otherwise* query that checks for a configurable condition and executes one action or another, depending on the condition.

Conclusions

The Shortcuts app is a useful tool for automating repetitive tasks in

everyday life and simplifying your work with macOS. Experienced admins and programmers who are used to the convenience of a full-blown development environment may be disappointed. For example, Shortcuts does not have a step debugger and does not return particularly meaningful error messages if something goes wrong. From an end user's point of view, Shortcuts nevertheless proves to be practical as long as the task is not too complex. My only regret is that not many third-party developers are looking to adapt their apps for Shortcuts right now. If you want more, you can use script actions to extend the functionality. ■

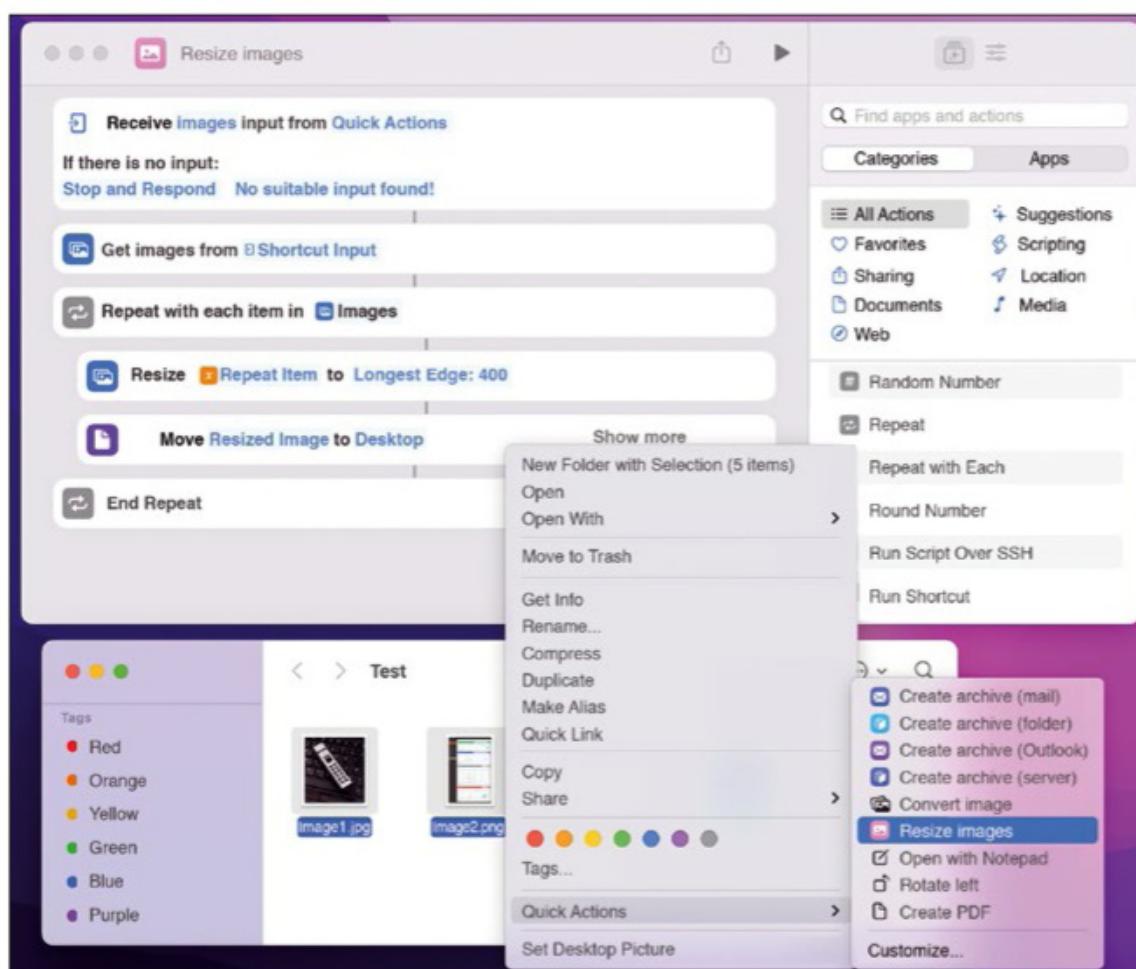


Figure 4: A loop runs actions against all objects passed in by the Finder.

Info

- [1] Intro to Shortcuts on Mac:
[<https://support.apple.com/en-gb/guide/shortcuts-mac/apdf22b0444c/5.0/mac/12.0>]
- [2] Importing Automator workflows:
[<https://support.apple.com/en-gb/guide/shortcuts-mac/apd02bffbaac/5.0/mac/12.0>]
- [3] Sharing shortcuts: [<https://support.apple.com/en-gb/guide/shortcuts-mac/apdf01f8c054/5.0/mac/12.0>]
- [4] Customizing the privacy settings:
[<https://support.apple.com/en-gb/guide/shortcuts-mac/apd961a4fc65/mac>]

Author

Christian Knerrmann is Head of IT-Management at Fraunhofer UMSICHT, a German research institute. He's written freelance about computing technology since 2006.



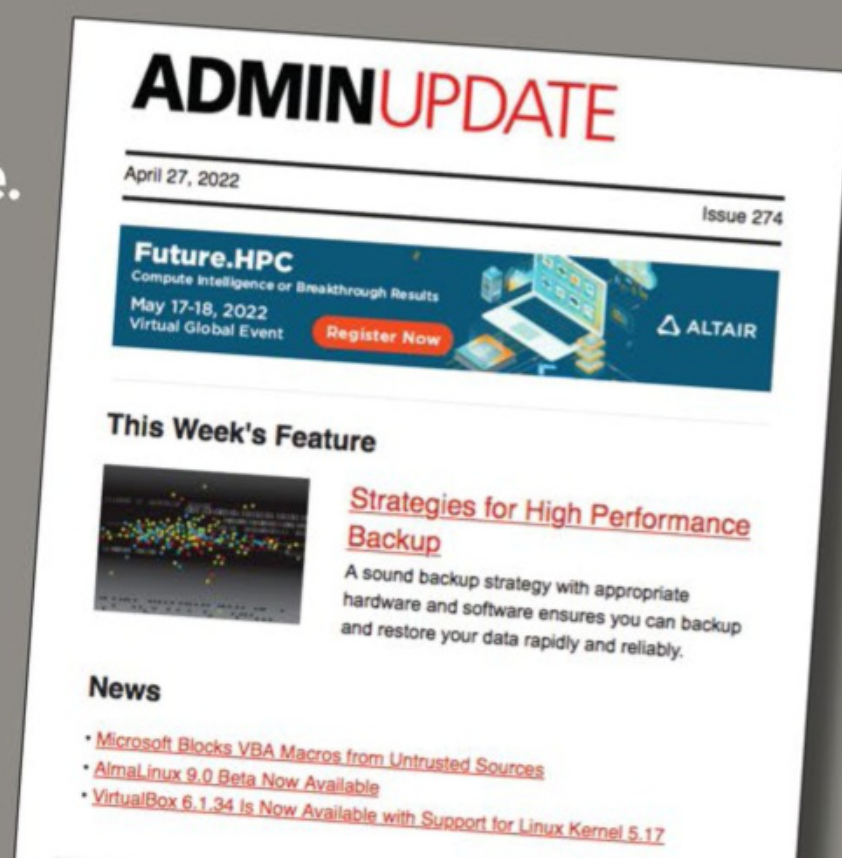
Too Swamped to Surf?



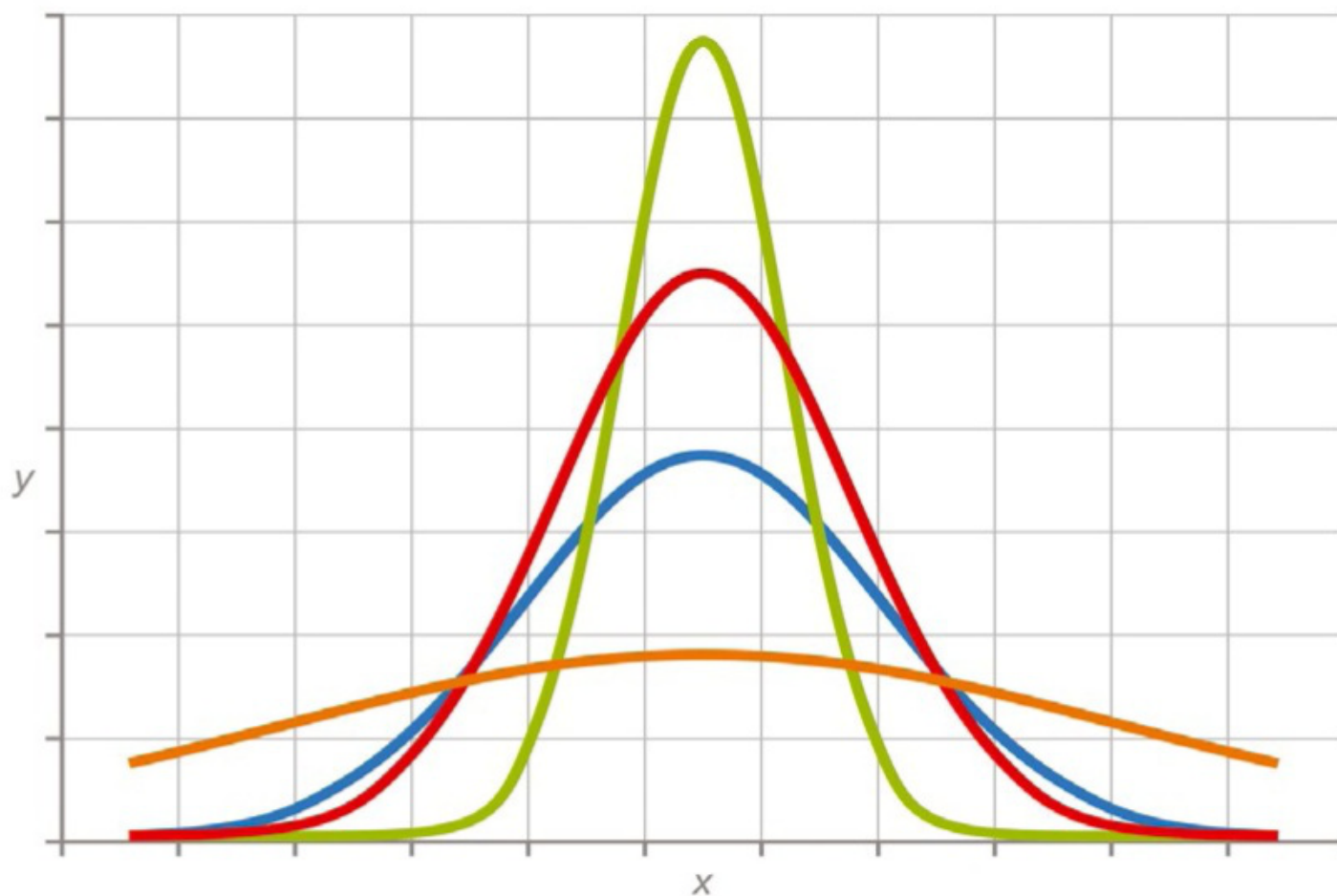
ADMIN offers additional news and technical articles you won't see in our print magazine.

Subscribe today to our free ADMIN Update newsletter and receive:

- Helpful updates on our best online features
- Timely discounts and special bonuses available only to newsletter readers
- Deep knowledge of the new IT



bit.ly/HPC-ADMIN-Update



Measuring the performance health of system nodes

Peak Performance

Many HPC systems check the state of a node before running an application, but not very many check that the performance of the node is acceptable before running the job. By Jeff Layton

In a previous article [1] I discussed prolog and epilog scripts with respect to a resource manager (job scheduler). Most of the prolog examples were simple and focused on setting up the environment before running a job, whereas the epilog example cleaned up after an application or workflow was executed. However, prolog and epilog scripts are not limited to these aspects. One aspect of prolog/epilog scripting that I didn't touch on was checking the health of the nodes assigned to the job.

Generically, you can think of a node health check as determining whether a node is configured as it should be (i.e., setting the environment as needed, which I discussed somewhat in the previous article) and is running as expected. This process includes checking that filesystems are mounted correctly, needed daemons are running, the amount of memory is correct, networking is up, and so on. I refer to this as the "state" of the node's health. In that same article, I mentioned Node Health Check (NHC) [2], which is used by several sites to check the health of nodes, hence the name. In my mind, it focuses on checking the "state" of the node, which is a

very important part of the health of a node. A large number of options can be turned on and off according to what you want to check about the state of the node.

Almost 20 years ago, when I worked for a Linux high-performance computing (HPC) company that no longer exists, we had a customer who really emphasized the number of nodes that were available for running jobs at any one time. One of the ways we measured this was to run a short application and check the performance against the other nodes. If the performance was up to or close to that of the other nodes, the node was considered "up" and available for users to run jobs. Otherwise, the node was considered down and not used to run jobs. I refer to this concept as the "performance" health of the node.

Performance-Healthy Nodes

Node performance health, at least in my mind, is greatly underappreciated. The node can be in a good state, where everything appears to be working, but it might not perform correctly (i.e., be healthy). The difficult question to answer is: What does "correctly" mean?

Because I'm talking HPC, the first typical reaction is to check that the node is running as fast as possible. Don't forget that parallel applications only run as fast as the slowest node. Having a few nodes that run faster than the others is fine (who doesn't like extra performance), but you want to find nodes that are running slower than most of the other nodes. To make this determination, you can check whether performance is less than the norm. In this article, I present some aspects for determining whether the performance of a node is acceptable for a user's application with the use of standard benchmarks that are run on the node before a user's application is run. The results are then compared with typical node performance for the same benchmarks.

Standard Benchmarks - NPB

The applications you run to check the health of a node are really up to you, but I highly recommend some typical applications. Although HPC runs multinode applications, an easy place to start is to focus on single-node application runs that can use all cores, all memory, or both. You can even run a

single core if you like. In my opinion, the emphasis should be on testing an important aspect of performance: peak CPU or GPU floating point operations per second (FLOPS), memory bandwidth, combinations of compute and memory, data transfer from the CPU to the GPU (bandwidth, latency, or both), data transfers to local or network storage, or scaling performance with more than one core.

An important aspect of the tests or benchmarks is that they don't take too much time to complete. You don't want to have a health check test that unduly delays starting a user's application, particularly for applications that might not run very long where a few minutes delay would be a big portion of the total runtime. You need a balance in the time it takes to complete the test, the usefulness of the test, and the user application execution time (which is likely to be unknown).

To determine performance, I use the NASA Advanced Supercomputing (NAS) parallel benchmarks [3], a set of code created by NASA to test performance of parallel systems, providing a proxy for computation and data movement in computation fluid dynamics (CFD) applications. NPB has been around for many years and has proven to be very useful in testing systems. One of the best aspects of the NPB tests is the different "classes" of problems that allow you to test a wide range of system configurations, which has a great effect on the time it takes to complete the tests. Over the years, NPB has added new classes of problem sizes, multizone versions of some tests (great for OpenMP coding), and even some tests that stress I/O and data movement.

In 1991, the NAS Division created some benchmarks [4] suited to their needs that had different problem sizes. So, they started writing NPB in 1991 and released it in 1992. Since then, NPB has had three major releases with new features, new code, and new languages.

The focus of the benchmarks was, as one would expect, derived from CFD applications. These benchmarks test many aspects of modern systems,

from stressing the CPU in different ways, to memory bandwidth aspects, to interprocess and network communications. NPB started with five "kernels" and three pseudo applications. The kernels are:

- IS – integer sort, random memory access
- EP – embarrassingly parallel
- CG – conjugate gradient, irregular memory access and communication
- MG – multigrid on a sequence of meshes, long- and short-distance communication, memory intensive
- FT – discrete 3D fast Fourier transform (FFT) emphasizing all-to-all communication

The three pseudo applications are:

- BT – block tri-diagonal solver
- SP – scalar penta-diagonal solver
- LU – lower-upper Gauss-Seidel solver

The latest version of NPB is 3.4.2. Before NPB version 2.3, the benchmarks were all serial unless the compiler created multiprocessor binaries. NPB version 2.3, released in 1997, comprised a complete version of the benchmarks that used the Message Passing Interface (MPI), although the serial versions were still available. In NPB release 3, three multizone versions of the benchmarks (termed NPB-MZ) were added that take advantage of MPI/OpenMP hybrid programming techniques and can stress interprocess communication and multinode communication, as well as single system aspects. A document [5] describing these benchmarks was released in 2003.

The multizone (MZ) benchmarks are:

- BT-MZ – uneven-size zones within a problem class, increased number of zones as the problem class grows
- SP-MZ – even-size zones within a problem class, increased number of zones as the problem class grows
- LU-MZ – even-size zones within a problem class, a fixed number of zones for all problem classes

NAS also created a set of benchmarks for unstructured computations, data I/O, and data movement:

- UA – unstructured adaptive mesh, dynamic, and irregular memory access

- BT-IO – a test of different parallel I/O techniques

- DC – data cube

- DT – data traffic

The UA computation benchmark can push random memory access and memory bandwidth very easily.

The primary changes made to the benchmarks of the latest version 3.4.2 of the NPB are:

- added class F to the existing S, W, A, B, C, D, E
- added dynamic memory allocation
- added MPI and OpenMP programming models

The latest version of the multizone benchmarks is also 3.4.2. The primary changes to these benchmarks are:

- codes: BT-MZ, SP-MZ, LU-MZ
- classes: S, W, A, B, C, D, E, F
- programming models: MPI + OpenMP, OpenMP
- added dynamic memory allocation

NAS has not released any official GPU versions of NPB, but if you look around the web, you can find various versions that others have created. Probably the most predominant version is NPB-GPU [6], which uses the CUDA parallel computing platform and programming model to produce the GPU versions of the five kernels and three pseudo applications.

Another common source of GPU versions is from Chemnitz University of Technology in the form of NPB-CUDA and NPB-MZ-CUDA [7].

Which Benchmarks to Use?

Before you can test node health, you have to establish baseline performance. Because the node tests will be run before a user's application, you don't want to spend too much time on the benchmarks. How much time you spend is really up to you, but for the sake of argument, I will limit the time to 30-40 seconds per benchmark.

Running all the NBP benchmarks would take too much time. Eight benchmarks at 30 seconds each is four minutes. I'm not sure how many users want to wait four minutes, before their applications start, to test that the nodes are running

correctly, but I can probably count them on one hand.

Instead, I'll run one or two benchmarks for a maximum of 30-80 seconds before the user's application runs.

For a new system, I run the various NPB benchmarks on a single node with the maximum number of cores possible. I measure the runtime with the Linux *time* command, and I run the benchmarks a few times to compute the mean (Table 1). For this article, I ran the tests on an AMD CPU with six non-threaded cores, most of them for classes A, B, and C (standard test problems, with an approximate four times size increase going from one class to the next) where possible.

Notice that for most benchmarks I could only use four cores. These benchmarks can only use a core count greater than one that is a multiple of two (i.e., 1, 2, 4, 8, 16, 32, etc.).

Looking through Table 1, I made the following observations and decisions:

- For my system, I could only run the IS test with a class C problem size. Therefore, I will restrict myself to classes A and B.
- The BT benchmark takes a while to run at class B sizes (66.5s), so I won't use that test.
- I want to have at least one test that uses all the cores, so that points to EP and LU. However, LU takes 41.8s for a class B size, which is a bit long, so I won't use the LU test, which leaves the EP test.
- The FT test is for a discrete 3D FFT with all-to-all communication. FFT can be communication intensive even on a single node.
- The total runtime of EP and FT for class B size is about 23s, which is a fairly short time, so I

will add the MG benchmark with a very short runtime (3.8s) for a class B problem size.

Therefore, I will run the EP, FT, and MG tests to check health performance. For class B, the EP test takes 5.46s, the FT test 17.26s, and the MB test 3.8s. If I stay with only class B tests, the total runtime would be about 27s, which I believe will be an acceptable time to wait before a job starts.

Acceptable Performance

The EP and FT tests measure performance. The next question to answer is: What is acceptable performance? The answer to this question will vary from site to site and from person to person, but I will discuss my ideas. You can't set acceptable performance by running benchmarks one time. You need to run them several times. For example, I would run each test a minimum of 21 times and collect the runtimes so that I can find any variation in the runs.

Next, I would compute the mean and standard deviation (SD) from the runtimes for each benchmark. To determine a cut-off value for performance, you should look at your

Table 1: NPB Test Results

Benchmark	Class A time (s)	Class B time (s)	Class C time (s)
BT (4 cores)	11.95	66.5	272
CG (4 cores)	0.4	23.9	62.3
EP (6 cores)	1.4	5.46	21.05
FT (4 cores)	1.69	17.26	67.7
IS (4 cores)	0.6	2.16	8.2
LU (6 cores)	5.13	41.8	
MG (4 cores)	1.2	3.8	39.1
SP (4 cores)	16.2	138.3	

distribution of runtimes. With a normal distribution (Figure 1), I don't care whether the performance is better than the mean, so I can ignore any performance cut-off to the right of the mean. (On the other hand, if the node performance is significantly better than the mean, you might want to try to understand why it is performing so much better than the others.)

The lower bound of performance, or the left-hand portion of Figure 1, is of most interest. As always in HPC, acceptable performance varies from site to site and from person to person, but I choose to cut off the lower limit of performance (LLP) at

$$\text{LLP} = \text{mean} - 1/2(\text{SD})$$

because 1 SD allows for a 34.1% reduction [8] in performance (assuming a normal distribution). One-eighth of a standard deviation is 4.78% of normally distributed data or 95.22% of the mean, allowing for benchmarks

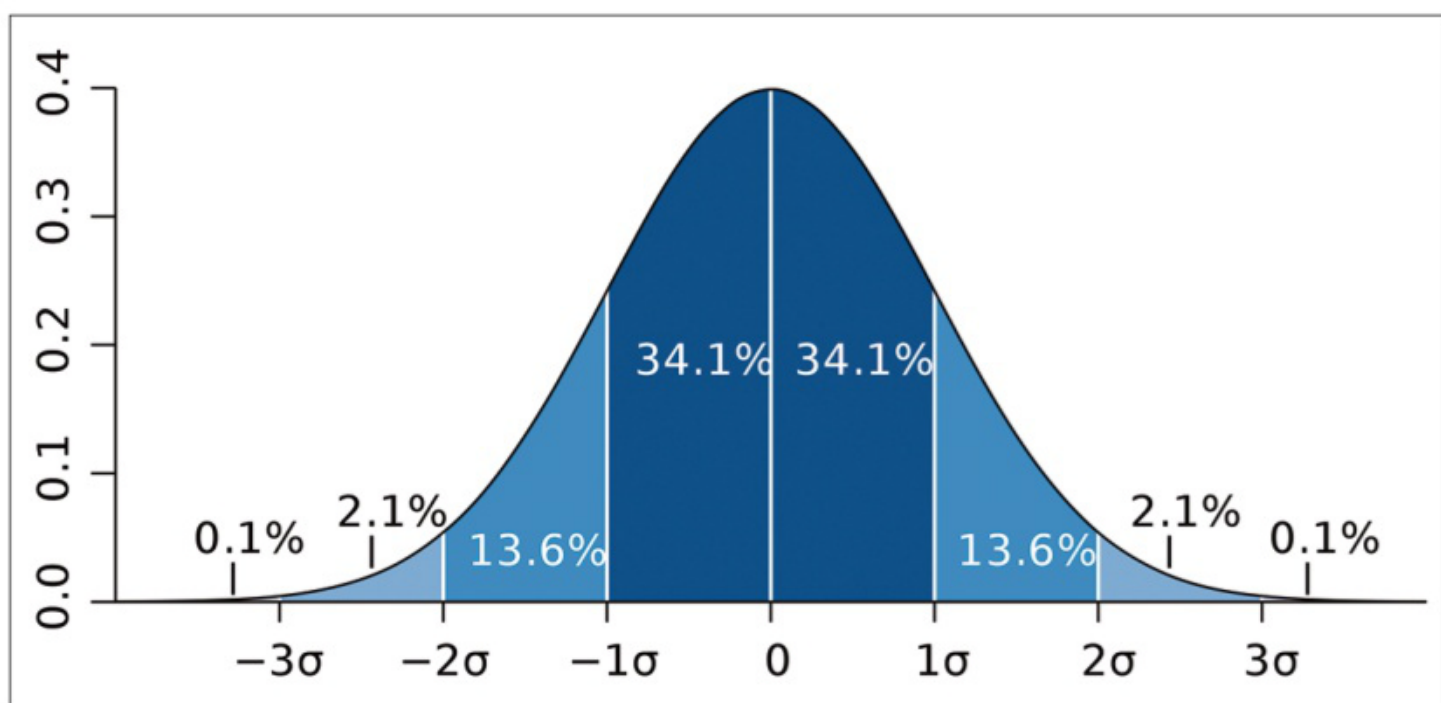


Figure 1: Normal distribution, with each band having a width of 1 standard deviation. Attribution 2.5 Generic (CC BY 2.5).

4.78% below the mean. If you select a standard deviation fraction of 0.01, then the minimum value is 96.02% of the mean.

To me, if I see more than a 5% drop in performance compared with the mean, I should probably start looking at the offending node. The problem could be noisy system processes, a user application that doesn't close correctly, or even incorrect DIMM or CPU replacements that don't match the speeds of the others.

Benchmark Process

Running health performance checks before a user application definitely sounds like the use of a prolog [1] in the resource manager, because prolog scripts run before a job. The prolog script can be almost anything: Just write the code as a Bash script, from which you can call scripts written in other languages such as Perl or Python, run applications, or execute Bash commands. Because prologs are run as root, be careful. For example, to run the EP benchmark use:

```
time mpirun -np 6 -h ./hostfile ep.B.x
```

The output could be written to a random file in /tmp and parsed for the runtime.

The benchmark time is then compared with the mean minus one-eighth of the standard deviation. If the test value falls below this computed value, the node is marked down, and the user's job is resubmitted. The benchmark runtime along with the node name is added to a file or database for unacceptable runtimes and the admin is notified. If the result is greater than the cutoff, you should write the benchmark time and node name to a file of acceptable results. This file should be used to recompute the mean and standard deviation for future benchmark testing. Four implicit assumptions underlie the process:

1. The benchmark binary and all supporting libraries and tools must be available on every node in the cluster on which you run jobs.

2. You need a script to parse the output from the benchmarks. Although not difficult, you have to write it, nonetheless.
3. You have to establish two files (databases): one for storing the unacceptable results and the other for storing acceptable results.
4. You need a resource manager that uses a prolog before running a job. All the HPC resource managers that I know of have this capability.

Summary

Checking the health of a node is an important task for maximum utilization. I like to divide node health into two parts: state health and performance health. Both parts can run in a prolog script in the resource manager before the user's application executes. These scripts can either report the results to the admin or be used to check that the node health is acceptable for running the user's application.

To determine the state health of the node, you can write scripts to check on the various states, such as whether filesystems are mounted correctly, needed daemons are running, the amount of memory is correct, networking is up, certain system packages have the required version, and so on. NHC is a great tool for checking the state of a node, with a wide range of tests. NHC is in production at several sites, some of which are very large. The tool has been well tested over several years and is easy to use. If you don't want to write your own prolog scripts, NHC is definitely the tool of choice.

The performance aspect of node health is often ignored in HPC system. I summarized some aspects of this health check with the use of standardized benchmarks. In particular, the NPB benchmarks were used as a basis for checking whether a node is performing well by testing various aspects of a system. These tests have been available since about 1992, are well understood, stress various aspects

of the system, and are very easy to compile.

By running the NPB benchmarks on the nodes of an HPC system several times, you can then determine the mean and standard deviation for the system. Then, you run the benchmarks on a node as part of a resource manager prolog to measure performance and compare the result with the mean and standard deviation of the system. If the results for the node are less than a desired value, the node is marked down and the user's job is resubmitted.

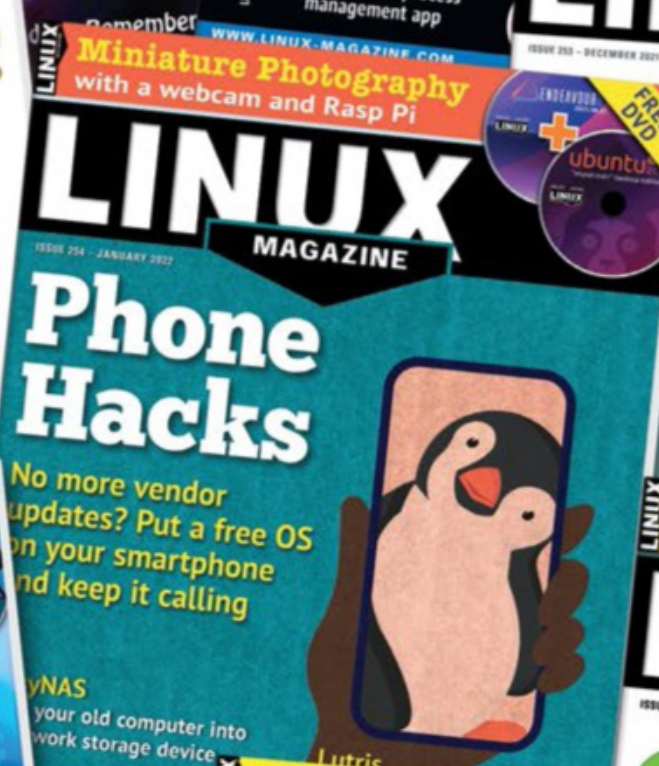
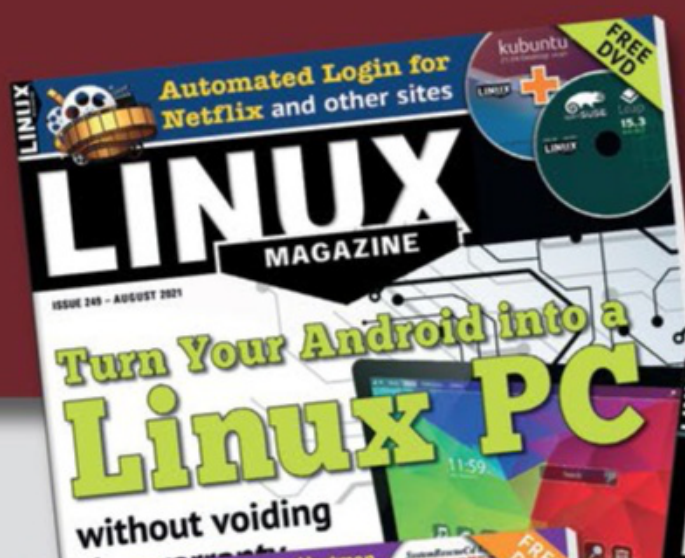
Determining a node's performance health before running a user's job can help both the user and the admin by reducing the number of questions around why an application might not be performing well. I hope this article has stimulated some ideas for measuring the performance health of your system. ■

Info

- [1] "Prolog and Epilog Scripts" by Jeff Layton: [\[https://www.admin-magazine.com/index.php/HPC/Articles/Prolog-and-Epilog-Scripts\]](https://www.admin-magazine.com/index.php/HPC/Articles/Prolog-and-Epilog-Scripts)
- [2] Node Health Check: [\[https://github.com/mej/nhc\]](https://github.com/mej/nhc)
- [3] NASA Advanced Supercomputing (NAS) Parallel Benchmarks: [\[https://www.nas.nasa.gov/software/npb.html\]](https://www.nas.nasa.gov/software/npb.html)
- [4] NAS Parallel Benchmarks Wikipedia: [\[https://en.wikipedia.org/wiki/NAS_Parallel_Benchmarks\]](https://en.wikipedia.org/wiki/NAS_Parallel_Benchmarks)
- [5] Benchmark document: [\[https://www.nas.nasa.gov/assets/pdf/techreports/2003/nas-03-010.pdf\]](https://www.nas.nasa.gov/assets/pdf/techreports/2003/nas-03-010.pdf)
- [6] NPB-GPU: [\[https://github.com/GMAP/NPB-GPU\]](https://github.com/GMAP/NPB-GPU)
- [7] NPB-CUDA and NPB-MZ-CUDA: [\[https://www.tu-chemnitz.de/informatik/PI/sonstiges/downloads/npb-gpu/index.php.en\]](https://www.tu-chemnitz.de/informatik/PI/sonstiges/downloads/npb-gpu/index.php.en)
- [8] 1SD Reduction: [\[https://www.mathsisfun.com/data/standard-normal-distribution-table.html\]](https://www.mathsisfun.com/data/standard-normal-distribution-table.html)

The Author

Jeff Layton has been in the HPC business for almost 25 years (starting when he was 4 years old). He can be found lounging around at a nearby Frys enjoying the coffee and waiting for sales.



Linux Magazine is your guide to the world of Linux. Look inside for advanced technical information you won't find anywhere else!

Expand your Linux skills with:

- In-depth articles on trending topics, including Bitcoin, ransomware, cloud computing, and more!
- How-tos and tutorials on useful tools that will save you time and protect your data
- Troubleshooting and optimization tips
- Insightful news on crucial developments in the world of open source
- Cool projects for Raspberry Pi, Arduino, and other maker-board systems

If you want to go farther and do more with Linux, subscribe today and never miss another issue!

Subscribe now!

shop.linuxnewmedia.com/subs

**GET IT
NOW!**

FAST DELIVERY
WITH OUR PDF
EDITION



Statistics and machine learning with Weka

One for All

The open source Weka tool applies a wide variety of analysis methods to data without the need for advanced programming skills and without having to change environments. By Manju Bhardwaj

Everyone has probably heard of machine learning, but how exactly does it work? Does it mean that an intelligent machine makes decisions on behalf of humans? In a way, yes, but strictly speaking, no. You might want to replace the term “intelligent machine” with “efficient algorithm” and add that this algorithm works with data. In doing so, it delivers a view that captures the essence of the data. Simply put, machine learning focuses on building models that learn from existing data and then uses those models to make logical decisions without requiring human intervention. The methods used to learn these models are the algorithms. A variety of algorithms exist, but no one of them is suitable for every case and everywhere. An algorithm that performs well on one data collection can fail on another, which is why researchers apply different algorithms to a given set of data to see which

algorithms work. If you had to program all these processes yourself, it would certainly be too difficult a task. That said, it is also tricky to find a platform that provides ready-made algorithms. Weka not only offers researchers a large number of ready-made machine learning algorithms, it also has features such as visualization and preprocessing.

Weka Basics

The open source Weka is licensed under the GNU General Public License (GPL) and was created at the University of Waikato in Hamilton, New Zealand. Interestingly, Weka is not an abbreviation or engineering

jargon. It is the name of a flightless bird that lives on the islands of New Zealand.

Written in Java, Weka runs on any operating system and hardware platform and is under constant development. Each update includes new features and ditches less popular ones. Version 3.8.6, released February 21, 2022, was current at the time this



Figure 1: The first window you encounter is the Weka GUI Chooser.

Photo by Shane Rounce on Unsplash

article was written. Version 3.9 was also available, but it might have contained some bugs, so I avoided it at the time. The uniqueness of the tool lies in the availability of a variety of different methods that cover almost all aspects of machine learning through a common interface.

The Weka wiki [1] holds a wealth of information and is where you will find installation instructions for Linux, macOS, and Windows. After the install, Weka can be used either directly in the Weka graphical user interface (GUI) or by application programming interface (API) calls in Java code. In this article, I only look at the GUI.

The first window you encounter is the Weka GUI Chooser (Figure 1), which you use to access one of five views:

- The *Explorer* tab lets you load the data you want to use and provides options for pre-processing and initial data analysis (Figure 2).
- Under *Experimenter*, you set up and run targeted experiments, analyze the results, and use it to compare the suitability of different methods for given data collections (Figure 3).
- *KnowledgeFlow* provides a graphical representation of the flow of information and a process-oriented view of the methods used. You can use this tool to design a machine learning pipeline from data input through results output, which you then run and evaluate in Weka.
- *Workbench* combines all the views already listed in one window (Figure 4).
- *SimpleCLI* is an alternative to the GUI that allows you to enter commands to control Weka.

Weka Explorer, the most widely used interface, provides tabs for a range of functions, including *Preprocess* (data preprocessing), *Classify* (classification algorithms), *Cluster* (clustering algorithms), *Associate* (association rules), *Select attributes* (feature selection), and *Visualize* (data visualization).

In the *Preprocess* tab you can load the data from a file, a database, or a link. Files are usually tabular in structure, consisting of columns and rows. Each

row represents an object (known as an instance in Weka), and each column represents a property of the object under investigation (an attribute). Weka supports real (floating-point values), integer, string, and nominal (i.e., yes and no) data types. Moreover, Weka supports a special extended CSV format known as the attribute-relation file format (ARFF). ARFF files have a header with information about the names and data types of the attributes.

After installing the Weka package, you will find example datasets in the `/usr/share/doc/weka/examples/` subfolder

Listing 1: diabetes.arff

```
@relation pima_diabetes
@attribute ,preg' numeric
@attribute ,plas' numeric
@attribute ,pres' numeric
@attribute ,skin' numeric
@attribute ,insu' numeric
@attribute ,mass' numeric
@attribute ,pedi' numeric
@attribute ,age' numeric
@attribute ,class' { tested_negative, tested_positive }
@data
6,148,72,35,0,33.6,0.627,50,tested_positive
1,85,66,29,0,26.6,0.351,31,tested_negative
8,183,64,0,0,23.3,0.672,32,tested_positive
1,89,66,23,94,28.1,0.167,21,tested_negative
[...]
```

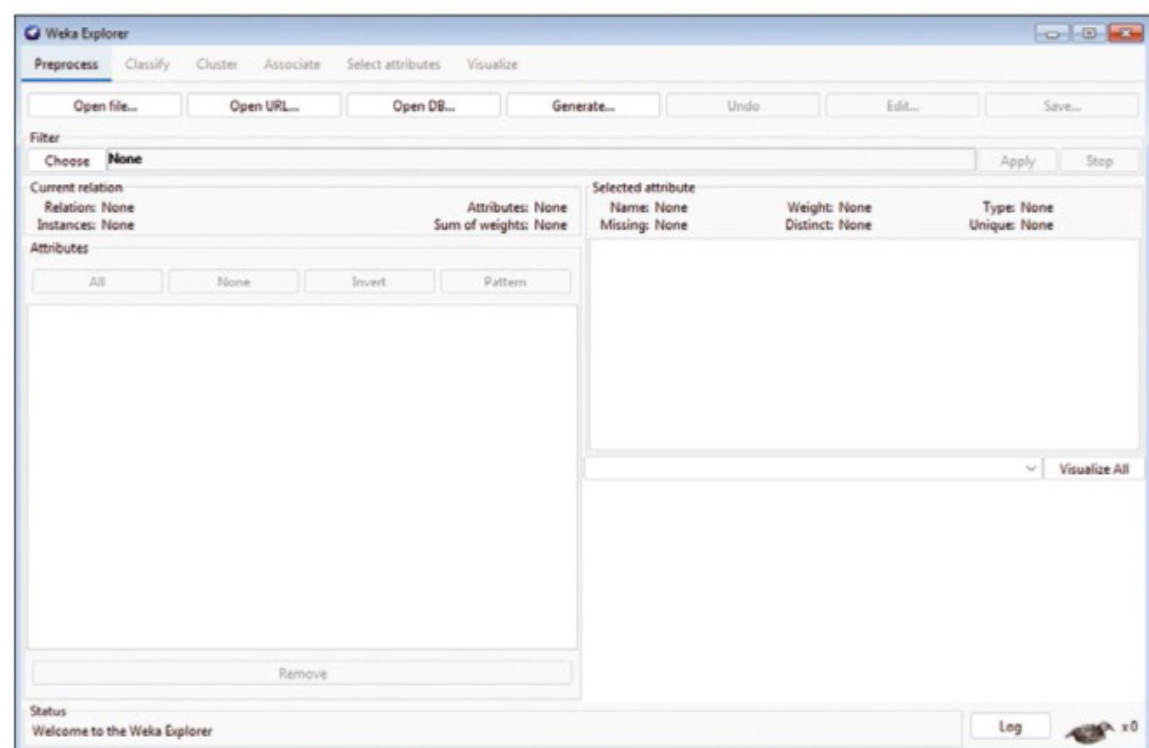


Figure 2: Weka Explorer lets you load the data and supports pre-processing and initial analysis.

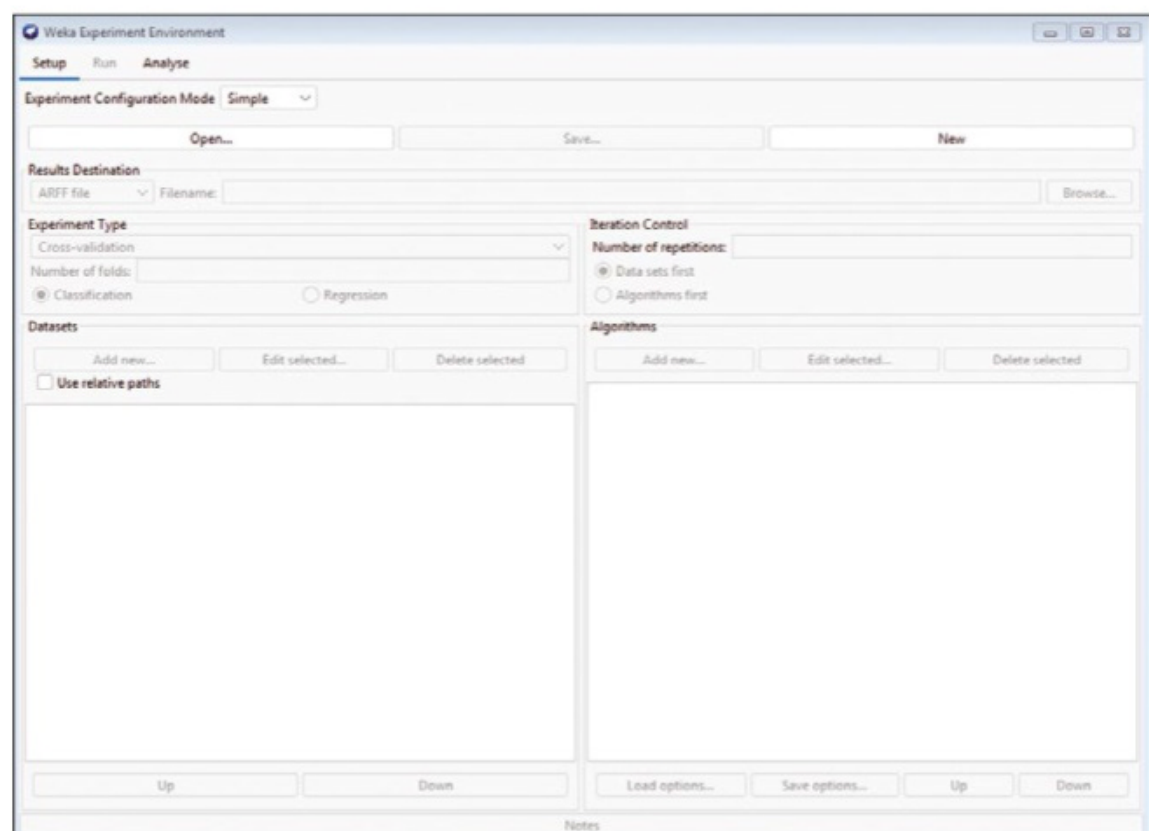


Figure 3: Weka Experiment Environment is suitable for comparing different methods of statistics or machine learning.

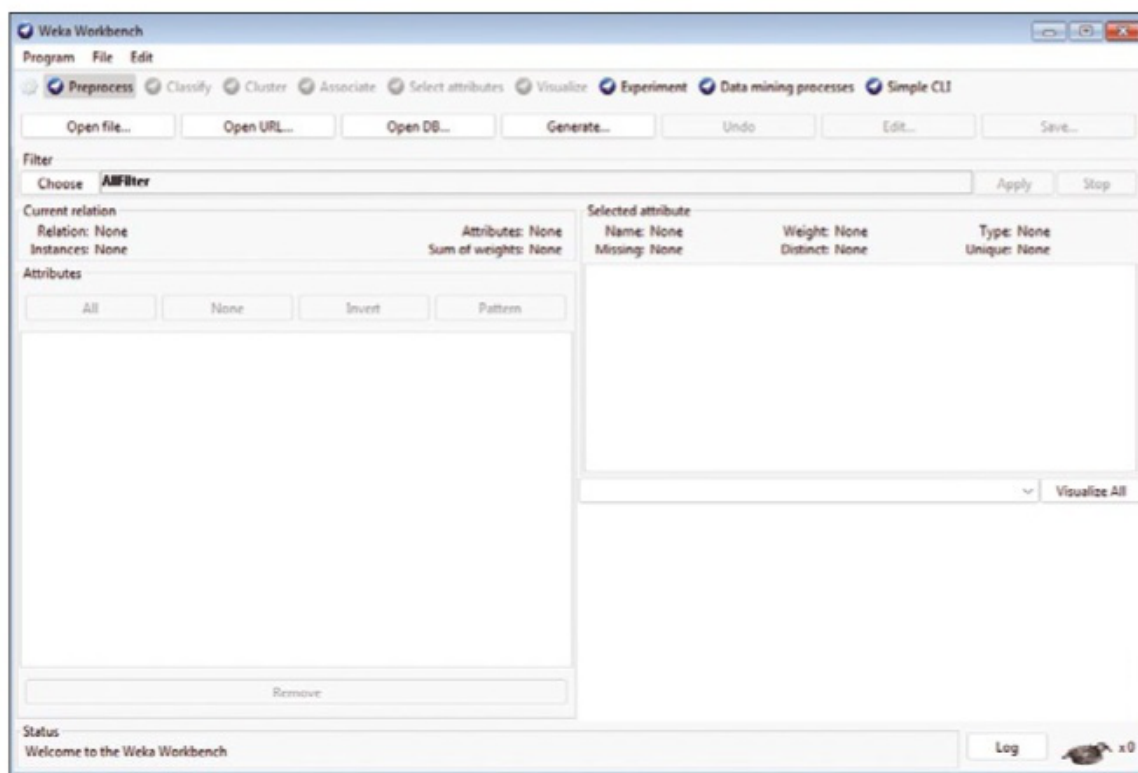


Figure 4: The Weka Workbench view puts everything in one window.

in the form of ARFF files that you can use to experiment with the tool. [Listing 1](#) shows an excerpt from `diabetes.arff`. To load this file in Explorer, press the *Open file* button.

You could load a CSV file in the same way after selecting that data type in the file dialog. However, this operation often results in problems that lead to error messages such as *Data*

values neither numeric nor nominal, which is why ARFF is the preferred file format for Weka.

CSV files can be converted to ARFF format by adding an appropriate header. To do this, first name the relation that the data file reflects (in this example, `@relation pima_diabetes`). Then, add information about all fields and their respective data types (e.g.,

`@attribute ,age' numeric`). Finally, add the class attribute – at least in this case. For applications that classify something, the attribute must be of the nominal type. To calculate a regression, you need the real data type. The data starts after the `@data` directive as comma-separated attribute values in rows.

After loading the file, Explorer displays some information about the data. In [Figure 5](#) you can see the number of instances and the names of the attributes on the left side. On the right side, Weka provides statistical key figures for the data (e.g., minimum, maximum, mean, etc.). Additionally, the tool shows a histogram for the attribute selected on the left.

Preprocessing

Before applying machine learning algorithms, you first need to clean up the data. The first step is to remove attributes that are not needed to analyze the problem. For example, the person's name would add nothing to the diabetes diagnosis. It

is best to remove such attributes to reduce the load for the algorithm and possibly improve performance at the same time. To do this, check the attribute boxes in question and click *Remove*. Weka provides a large number of filters that you can apply to the dataset. For example, if you are only interested in certain values of an attribute, you can hide all others with a filter. Another example would be the desire to normalize certain variables

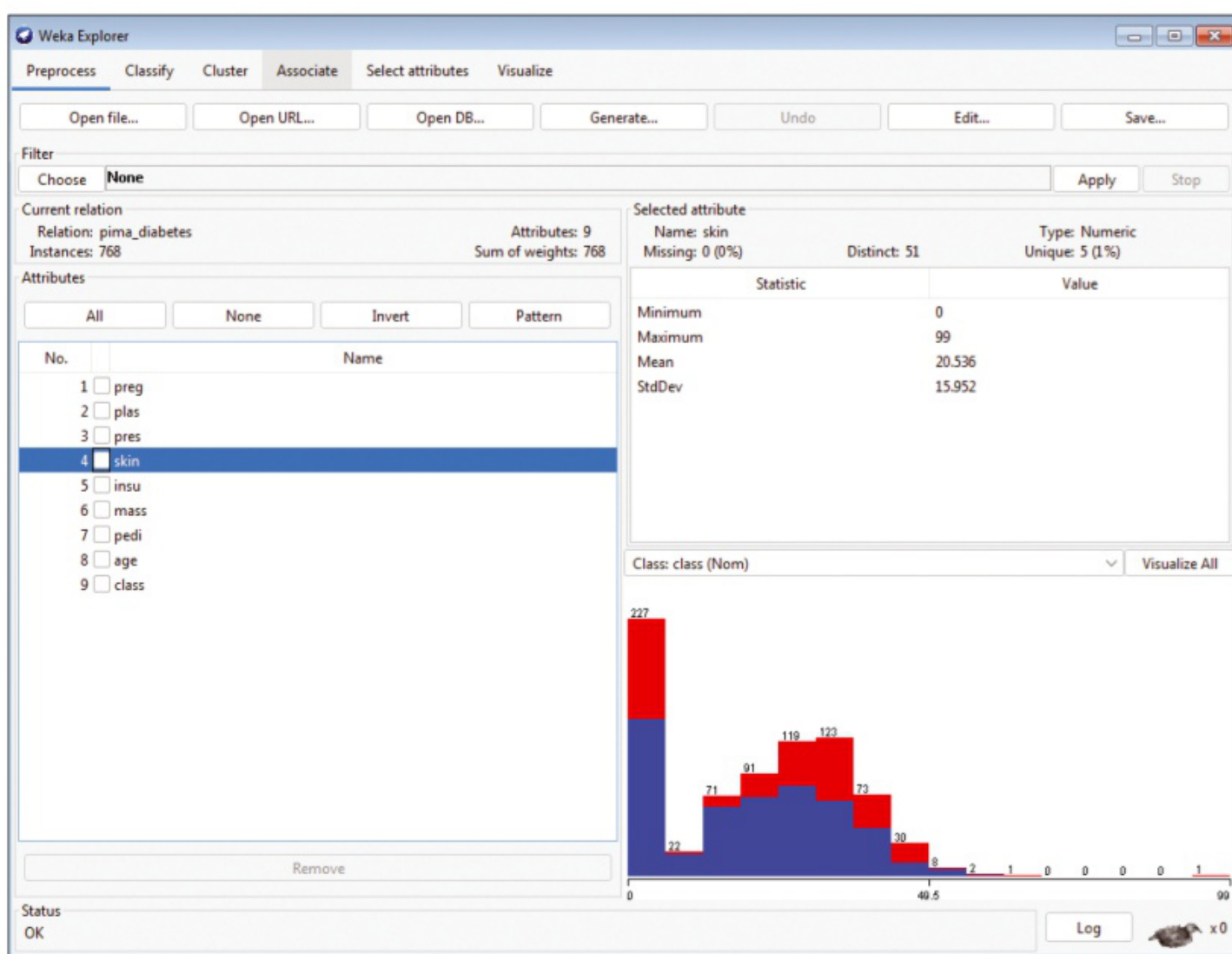


Figure 5: The diabetes sample data after loading into Weka Explorer.

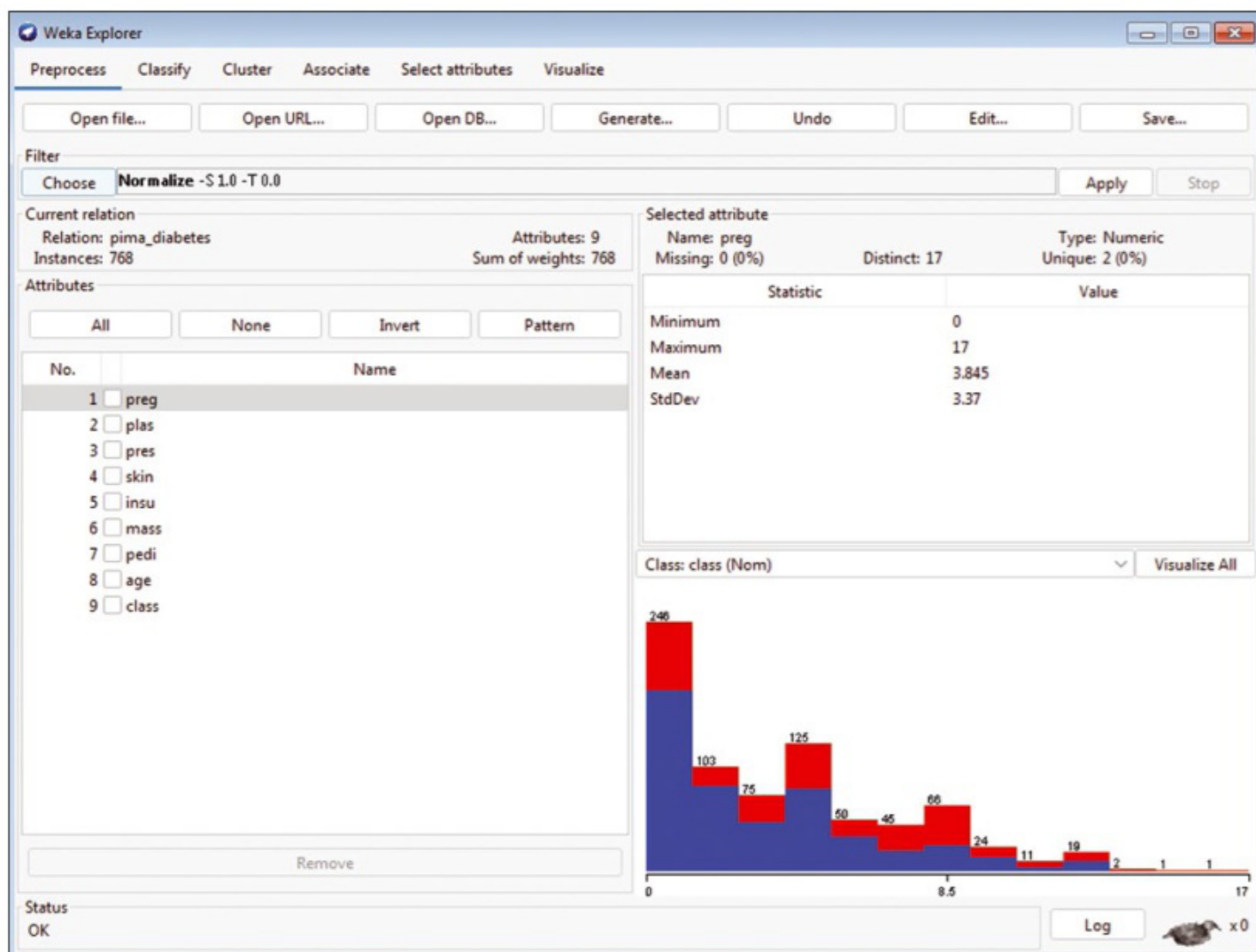


Figure 6: A normalization filter applied to an attribute of the sample dataset.

such as age or income. In any case, this step requires an understanding of both the data and the algorithm to decide which filters to consider. You can select filters by clicking the *Choose* button below the Filter label. Two types of filters are available: supervised and unsupervised filters. Supervised filters use class values and are very rare compared with unsupervised filters. A distinction is also made between class and instance filters. The *Allfilter* or *Multifilter* options let you combine filters.

Figure 6 shows how a normalization filter was applied to the *preg* attribute of the sample dataset. The text box to the right of the *Choose* button contains the parameters for the filter. By the way, the Explorer also has an *Undo* button, which can be used to undo any change, and a *Save* button to save the current values to a file.

Machine Learning

After preprocessing, you can move on to an arbitrary machine learning task, be it classification, regression calculation, clustering, or association rule

discovery. The *Classify* tab provides an interface to many very well known classification algorithms, including decision trees, support vector machines, naive Bayes, multilayer perceptrons, and logistic regressions, to name a few. Meta-learning with bagging, boosting, and stacking is also possible with Weka. Cross-validation

version of C4.5 programmed in Java; it generates a decision tree classifier. In Explorer, first select *Classify* from the top menu bar and then press the *Choose* button to select J48 from the list of classification algorithms. The algorithm and some default parameters will appear in the text box to the right (**Figure 7**). You can edit the

and the hold-out method are available to evaluate the classifier. The parameters of all of the algorithms can be adjusted. Metrics such as accuracy, precision, or recognition value can be used to evaluate the predictive performance of a classifier.

The following example is intended to show how a decision tree classifier can be applied to the sample data set diabetes.

arff. The J48 algorithm is used, which is the open source ver-

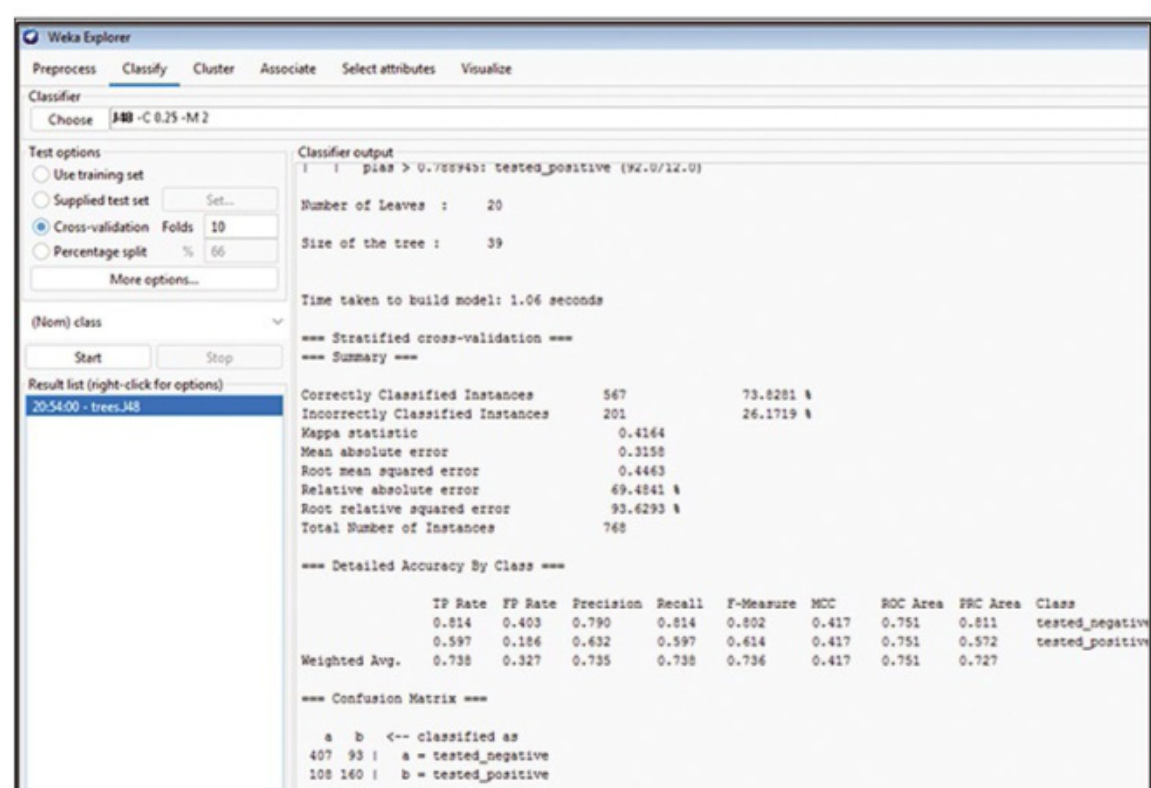


Figure 7: The Classifier window is used to configure and apply the J48 decision tree algorithm.

parameters at any time, provided you have the necessary understanding of the algorithm's function and the meaning of its parameters.

The classifier can use the data in several ways. Either it treats the entire data set as a training set and generates a model that can be saved and reloaded later to predict new test data, or you can use the data in a cross-validation or hold-out method and split the data into a training set and a test set on a percentage basis. This method is usually employed to evaluate the performance of the classifier for the given data set.

If you click on *More options*, you can select the evaluation metrics to use, along with some other options. However, this step, which you can also skip at the beginning, is optional. Next, press the *Start* button to run the desired task with the

selected classifier. The right part of the window displays the steps taken by the classifier and the results with the values of the selected metrics. You now have the option to run the same classifier over and over again with the same or different parameters or options.

If the performance of one classifier does not satisfy you, simply switch to another. In this way, you can try out different algorithms with the same tool in a graphical interface to find the one best suited to your use case. If you want to focus on a single classifier, you can automatically tune the parameters by cross-validation to determine the optimal values. Following exactly the same approach, you can also use cluster algorithms like SimpleKMeans, FarthestFirst, or HierarchicalClusterer in the corresponding tab. Association rule mining can be performed from the *Associate* tab, and

the Apriori algorithm helps mine frequent patterns.

Finally, the *Visualize* tab helps with data visualization. For example, scatter plots of all attribute combinations can be output automatically (**Figure 8**). The different colors represent different class memberships, which is very helpful in discovering relationships between attributes and, in turn, assists in filtering and analysis.

Conclusions

Weka can help you simplify your machine learning tasks. Although it is impossible to do justice to Weka in a short article, the strength of the tool lies in its practical application. The more familiar the terms and tasks of machine learning become, the better you will get along with Weka. Newcomers to machine learning will certainly appreciate the ease and flexibility that Weka offers. The tool prevents researchers from getting bogged down with difficult programming tasks. I have used Weka extensively in both teaching and research, and I highly recommend it to anyone. ■

Info

[1] Weka wiki:

[<https://waikato.github.io/weka-wiki/>]

The Author

Manju Bhardwaj is Associate Professor at the Department of Computer Science, Maitreyi College, University of Delhi, and has more than 25 years of teaching experience. Her doctoral thesis focused on classification ensembles. She has participated in and presented papers at numerous conferences and published articles in prestigious international journals. She is also an active reviewer of IEEE journals.

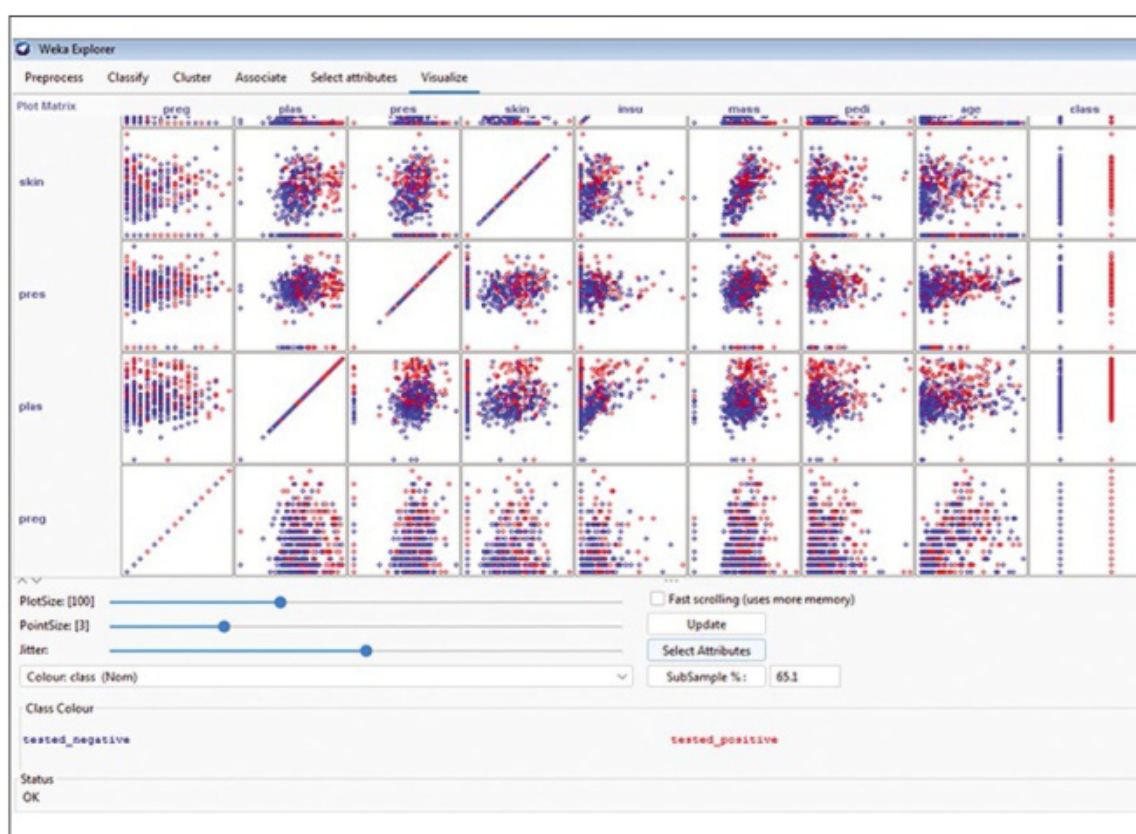


Figure 8: The *Visualize* tab lets you examine the data visually – for example, as a scatter plot, as shown here.

Public Money

Public Code



Modernising Public Infrastructure with Free Software

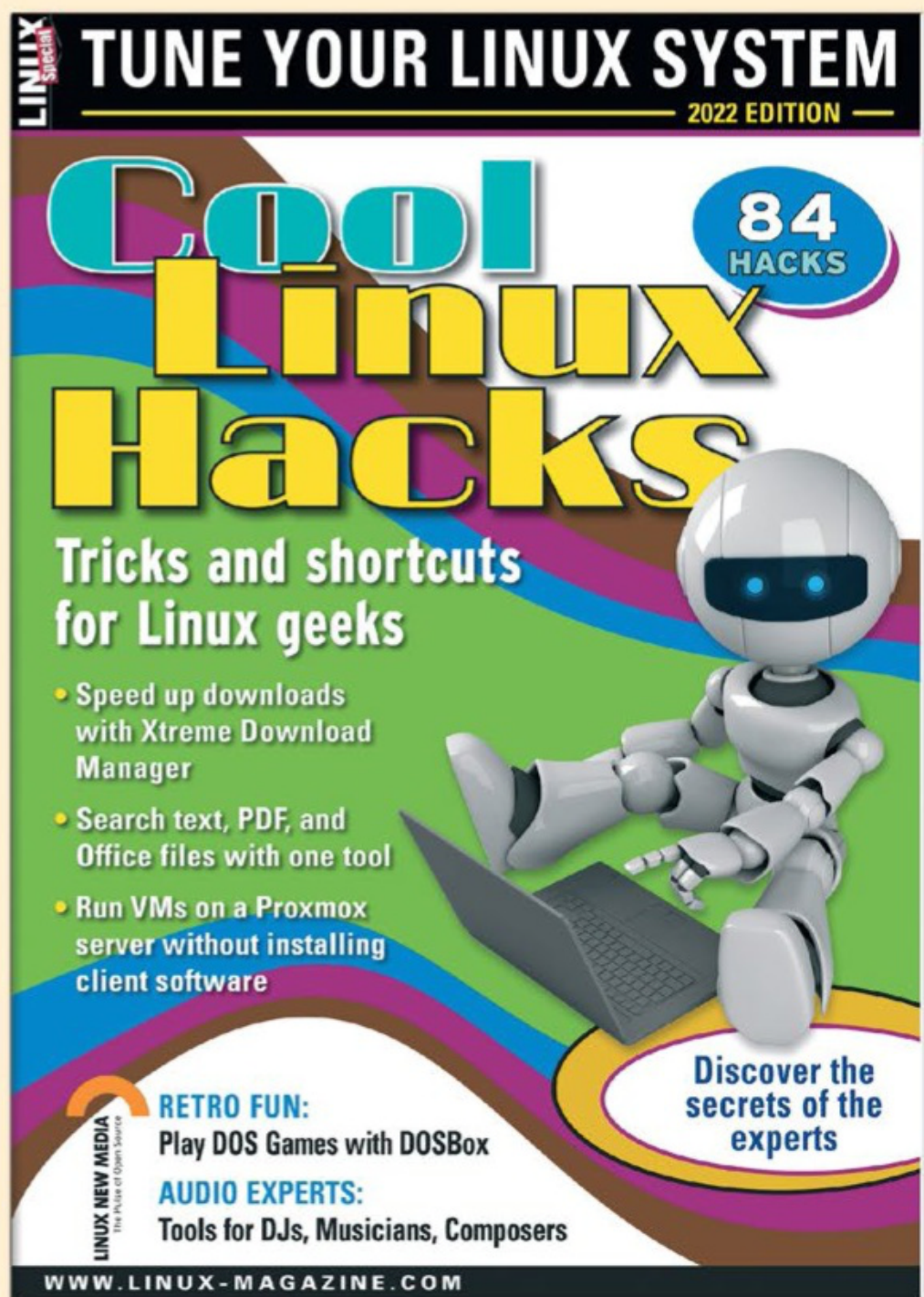


Free Software Foundation Europe

Learn More: <https://publiccode.eu/>

SHOP THE SHOP
shop.linuxnewmedia.com

GET PRODUCTIVE WITH COOL LINUX HACKS



Improve your Linux skills with this cool collection of inspirational tricks and shortcuts for Linux geeks.

- Google on the Command Line
- OpenSnitch Application Firewall
- Parse the systemd journal
- Control Git with lazygit
- Run Old DOS Games with DOSBox
- And more!

ORDER ONLINE:
shop.linuxnewmedia.com/specials

ADMIN

Network & Security

NEWSSTAND

Order online:
bit.ly/ADMIN-Newsstand

ADMIN is your source for technical solutions to real-world problems. Every issue is packed with practical articles on the topics you need, such as: security, cloud computing, DevOps, HPC, storage, and more! Explore our full catalog of back issues for specific topics or to complete your collection.

#69 - May/June 2022

Terraform

After nearly 10 years of work on Terraform, the HashiCorp team delivers the 1.0 version of the cloud automation tool.

On the DVD: ubuntu 22.04 "Jammy Jellyfish" LTS server Edition



#68 - March/April 2022

Automation in the Enterprise

Automation in the enterprise extends to remote maintenance, cloud orchestration, and network hardware

On the DVD: AlmaLinux 8.5 (minimal)



#67 - January/February 2022

systemd Security

This issue, we look at how to secure systemd services and its associated components.

On the DVD: Fedora 35 Server (Install)

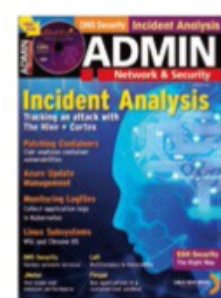


#66 - November/December 2021

Incident Analysis

We look at updating, patching, and log monitoring container apps and explore The Hive + Cortex optimization.

On the DVD: Ubuntu 21.10 "Impish Indri" Server Edition

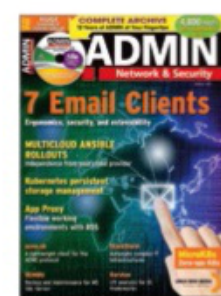


#65 - September/October 2021

7 Email Clients

The features in this issue tackle digital certificates, email clients, and HP backup strategies.

On the DVD: Complete ADMIN Archive DVD



#64 - July/August 2021

Bare Metal Deployment

Setting up, automating, and managing bare metal deployments gets easier with the tools presented in this issue.

On the DVD: Rocky Linux 8.4 (Minimal Install)



WRITE FOR US

Admin: Network and Security is looking for good, practical articles on system administration topics. We love to hear from IT professionals who have discovered innovative tools or techniques for solving real-world problems.

Tell us about your favorite:

- interoperability solutions
- practical tools for cloud environments
- security problems and how you solved them
- ingenious custom scripts

- unheralded open source utilities
 - Windows networking techniques that aren't explained (or aren't explained well) in the standard documentation.
- We need concrete, fully developed solutions: installation steps, configuration files, examples – we are looking for a complete discussion, not just a “hot tip” that leaves the details to the reader.

If you have an idea for an article, send a 1-2 paragraph proposal describing your topic to: edit@admin-magazine.com.



Authors

Amber Ankerholz	8
Dr. Manju Bhardwaj	90
Thomas Drilling	50
Steffen Eid	20
Florian Frommherz	68
Trevor Grant	22
Donnie Greer	32
Ken Hess	3
Thomas Joos	72
Christian Knerrmann	78
Martin Kuppinger	10
Jeff Layton	84
Martin Gerhard Loschwitz	26, 42, 56, 62
Andrew Musselman	22
Dr. Holger Reibold	46
Andreas Stolzenberger	36
Matthias Wübbeling	16

Contact Info

Editor in Chief

Joe Casad, jcasad@linuxnewmedia.com

Managing Editors

Rita L Sooby, rsooby@linuxnewmedia.com
Lori White, lwhite@linuxnewmedia.com

Senior Editor

Ken Hess

Localization & Translation

Ian Travis

News Editor

Amber Ankerholz

Copy Editors

Amy Pettie, Aubrey Vaughn

Layout

Dena Friesen, Lori White

Cover Design

Dena Friesen, Illustration based on graphics by
Oleksandr Omelchenko, 123RF.com

Advertising

Brian Osborn, bosborn@linuxnewmedia.com
phone +49 8093 7679420

Publisher

Brian Osborn

Marketing Communications

Gwen Clark, gclark@linuxnewmedia.com
Linux New Media USA, LLC
4840 Bob Billings Parkway, Ste 104
Lawrence, KS 66049 USA

Customer Service / Subscription

For USA and Canada:
Email: cs@linuxnewmedia.com
Phone: 1-866-247-2802
(Toll Free from the US and Canada)

For all other countries:
Email: subs@linuxnewmedia.com
www.admin-magazine.com

While every care has been taken in the content of the magazine, the publishers cannot be held responsible for the accuracy of the information contained within it or any consequences arising from the use of it. The use of the DVD provided with the magazine or any material provided on it is at your own risk.

Copyright and Trademarks © 2022 Linux New Media USA, LLC.

No material may be reproduced in any form whatsoever in whole or in part without the written permission of the publishers. It is assumed that all correspondence sent, for example, letters, email, faxes, photographs, articles, drawings, are supplied for publication or license to third parties on a non-exclusive worldwide basis by Linux New Media unless otherwise stated in writing.

All brand or product names are trademarks of their respective owners. Contact us if we haven't credited your copyright; we will always correct any oversight.

Printed in Nuremberg, Germany by Zeitfracht GmbH.

Distributed by Seymour Distribution Ltd, United Kingdom

ADMIN is published bimonthly by Linux New Media USA, LLC, 4840 Bob Billings Parkway, Ste 104, Lawrence, KS 66049, USA (Print ISSN: 2045-0702, Online ISSN: 2831-9583). July/August 2022.

Periodicals Postage paid at Lawrence, KS. Ride-Along Enclosed. POSTMASTER: Please send address changes to ADMIN, 4840 Bob Billings Parkway, Ste 104, Lawrence, KS 66049, USA.

Represented in Europe and other territories by: Sparkhaus Media GmbH, Bialasstr. 1a, 85625 Glonn, Germany.



DrupalCon
PRAGUE2022

JOIN US!

Just a few weeks to go before

DRUPALCON PRAGUE 2022

20 - 23 SEPTEMBER

Visit our website <https://events.drupal.org/prague2022> for more information.
Do you have any question? Contact us any time at drupal@kuonitumlare.com



**Don't forget to register
before it's too late!**



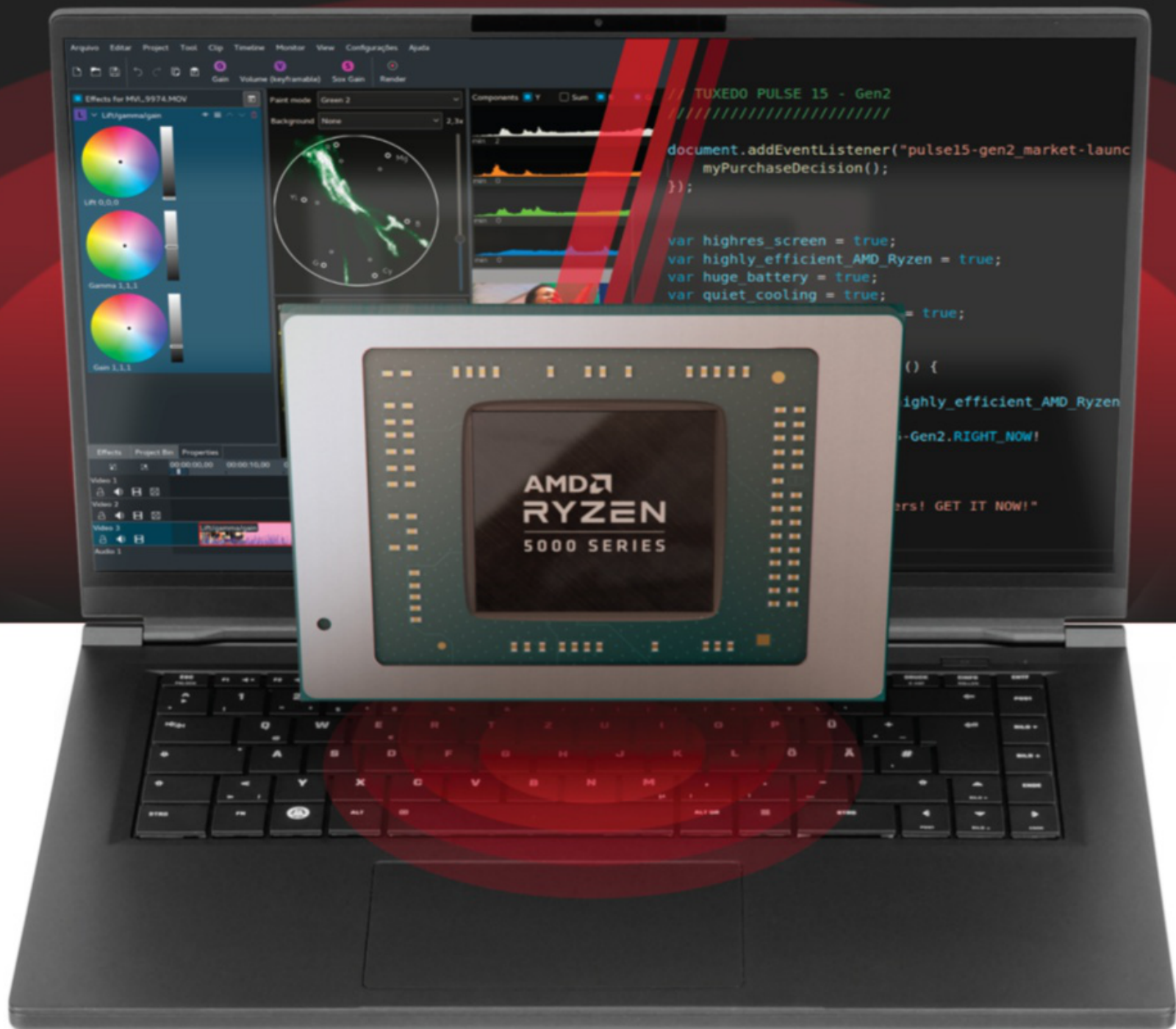
**Take a look
at the Program**

**Explore sponsorship
opportunities**

**Become a Volunteer at
DrupalCon Prague 2022**

**Plan your trip
to Prague**





Qui(e)te powerful!

TUXEDO Pulse 15 - Gen2



AMD Ryzen 7 5700U-35W
8 cores | 16 threads



WQHD display
2560 x 1440 | 165 Hz



Up to 18 h of runtime
91 Wh battery



Rigid magnesium chassis
1,7 cm thin | 1,5 kg light



100%
Linux

5

Year
Warranty



Lifetime
Support



Built in
Germany



German
Privacy



Local
Support

TUXEDO 18th
COMPUTERS ANNIVERSARY

tuxedocomputers.com

*apply
now!*

Looking for a new job?
tuxedocomputers.com/jobs