

# ADMIN

## Network & Security

ISSUE 80

# Threat MANAGEMENT

### Lock down your IT environment

#### Security Onion

Intrusion detection  
on the network

#### 7 Tools for container monitoring

#### Velociraptor

Detecting cyber threats in  
industrial environments

Basic strategies  
to secure mail  
servers

#### Azure Application Gateway

Managed service for  
Layer 7 load balancing

#### MySQL Migration

Manage breaking changes in  
MySQL 5.7 to 8.0 upgrade

#### DefectDojo

Fix vulnerabilities early in  
the development cycle

#### Kubescape

Kubernetes container  
security and compliance



## Why TUXEDO? Hardware. Software. Service.



Our devices come pre-installed with TUXEDO OS and are perfectly optimized for use with Linux.



Our software solutions are available to all Linux users and are continuously developed.



TUXEDO's technical customer service is always available for questions, requests and assistance.



Linux  
compatible



Up to 5  
Years Guarantee



Immediately  
ready for use

# TUXEDO

[tuxedocomputers.com](https://tuxedocomputers.com)



Made in  
Germany



German Data  
Privacy



German  
Tech Support



# Gleaning Agile Methodology for Infrastructure Projects

I remember when this whole agile methodology (Agile) thing hit several years ago. Project managers began spouting crazy new terms such as sprints, scrum, waterfall, and stand-ups. I'm not a fan of corporate buzzspeak, but Agile has brought a whole new hellscape into focus for those involved in so-called operations. Operations groups include system administrators, database administrators, cloud engineers, network administrators, and the like. We are the people who support systems and services. We are the ones who keep things going. We have hands on keyboards and are the folks in the trenches. In corporate speak, we are responsible for business continuity and day-to-day operations. The last thing we need is another pointless meeting – especially a daily meeting.

Agile might work well for managing the rollout of a new application, codebase, or something related to software development. Keeping close tabs on developers and their progress toward hitting deadlines is wise, but for operations, it's another pain we must endure as lowly tech workers. We have real-time chat services such as Microsoft Teams or Slack that keep us in continuous contact with coworkers. Ours are days driven by interrupts and constant collaboration. The need for a formal meeting to track daily progress is as useful as an intravenous drip of bubonic plague.

For operations groups, Agile is disruptive, and it causes important support issues to be missed because of the daily pressures of having to present a line-item list of what we've done, what we're going to do, and anything that might block us in performing those functions. FYI – my greatest block to productive success is the daily stand-up meeting, scrum call, or whatever you call it. If I'm in the middle of editing configuration files on multiple systems, I don't want a meeting to fall during my work. My connections time out. I forget what I've done and where. My momentum is lost. And for what? A meeting to tell everyone what they already know from watching their chat clients.

The many disadvantages outweigh any advantages Agile might confer. One of the glaring disadvantages is that success is difficult to measure. Reaching an arbitrary milestone is not a measure of success for operations personnel. We measure success by having fewer tickets to manage. Happy systems and happy users are our yardsticks, not calendar dates or lists of completed tasks. In support roles, a project has no clear destination or finish. Support is ongoing, such as patching, upgrading, performance tweaking, adding capacity, monitoring, user support, backups, and security. Therefore, the Agile project mentality doesn't work for our projects. Executive managers who take classes, attend seminars, or read about some new trend and then force it upon the unwashed masses would never acknowledge it, but they diminish productivity by adopting trendy programs such as Agile into the workplace.

Still, the Agile trend persists in operations. The old “beat to fit and paint to match” saying applies to the attempt to create a complete Agile workplace. It's a failure, but I get it. I understand the idea behind Agile for infrastructure operations projects. Your management wants an Agile workplace, and that's final; even if it doesn't work, you must make it work. I'm sure the most often quoted reason for implementing Agile in infrastructure projects is, “Hey, it works at Company X, so we're going to do it.” I'll bet if you ask the good people at Company X, they'll tell you that it, in fact, does not work for them either. They tolerate it because they must.

Agile has one positive outcome, and that is it does keep infrastructure teams focused on the tasks at hand. Managing IT people is often compared with herding cats, and Agile helps harness energies into maintaining a course rather than have everyone go off on their own tangential paths and leave

critical tasks half done; so, it isn't all bad. We should extract the positives from Agile and use those to move our projects forward. The projects and goals from Agile meeting notes could easily be incorporated into the very useful kanban board where progress is visual rather than captured in meeting notes that no one reads. Trello, a web-based, kanban-style application, works quite well for infrastructure teams to maintain focus and proceed through tasks.

All in all, Agile methodologies have their place, but not in infrastructure projects. We should glean the positives from it and leave the rest behind. Sorry to cut this short, but it's time for my scrum meeting. See you next time.

Ken Hess • Senior ADMIN Editor

# ADMIN

## Network & Security

### Features

- 10 Threat Detection**  
The open source tool Velociraptor is at the heart of a solution that automatically detects cyber threats in industrial environments, offering a defensive strategy and protecting critical infrastructures.
- 16 Container Security**  
Vulnerable software and incorrect settings cause problems, but administrators have some tools at hand that can help with container security.
- 22 Hardening Mail Servers**  
If you don't want to hand over your mail to a corporation, you have to operate your own mail server. Securing it sensibly involves some effort.
- 28 Security Onion**  
This network and enterprise security monitoring solution bundles numerous individual Linux tools that help you monitor networks or fend off attacks to create a standardized platform for securing IT environments.

### Service

- 3 Welcome**
- 6 News**
- 97 Back Issues**
- 98 Call for Papers**

### Tools

- 32 Azure Application Gateway**  
In the Azure cloud, Microsoft offers the AAG managed service as a Layer 7 load balancer that needs virtually no internal resources to set up and operate.
- 36 Scripting Chatbots**  
We look at the limited capabilities of artificial intelligence scripting with free large language models and suggest where it might work best.
- 40 MySQL 5.7 to 8.0 Upgrade**  
A number of breaking changes have been introduced between MySQL 5.7 and 8.0. We show you how to navigate this mandatory upgrade.

### Containers and Virtualization

- 46 Compact Kubernetes**  
We look at three scaled-down, compact Kubernetes distributions for operation on edge devices or in small branch office environments.
- 52 Microsoft Bicep**  
This fairly new domain-specific language creates Azure resources in the cloud. What can it do, and what advantages does it offer admins?
- 56 WPCloudDeploy**  
Reduce hosting costs for your fleet of WordPress cloud servers by managing dozens or even hundreds of servers and sites with WordPress-specific smarts.



@adminmagazine



@adminmag



ADMIN magazine



@adminmagazine



# 10 | Threat Management

## Lock down your IT environment

Digital infrastructures are vulnerable to all kinds of attacks. You need strategies and tools to detect and defend.

### Highlights

#### 10 Threat Detection

Velociraptor lets you monitor, analyze, and save data sources in the programmable logic controller environment for adaptive detection of threats on CRITIS networks.

#### 40 MySQL Migration

Changes implemented between MySQL 5.7 and the 8.0 release could pose significant obstacles, but with a little foresight, you can confidently and successfully pull off your upgrade.

#### 46 Compact Kubernetes

Kubernetes is starting to find its way into branch offices and small edge installations, which has prompted the appearance of MicroShift, MicroK8s, and K3s lightweight Kubernetes distributions.

### Security

#### 66 DefectDojo

This vulnerability management tool helps development teams and admins identify, track, and fix vulnerabilities early in the software development process.

#### 70 Windows Ransomware Protection

Ransomware defense involves two strategies: identifying attacks and slowing the attackers to mitigate their effects.

#### 76 Kubescape

Scan Kubernetes container setups for vulnerabilities and misconfigurations to improve security and compliance.

### Management

#### 82 RustDesk

An open source client and basic server are the basis for this self-hosted, cross-platform remote desktop access software designed to provide support and maintenance.

### Nuts and Bolts

#### 88 Hyper-V Resilience

Back up Hyper-V with Azure Site Recovery and prepare for failover.

#### 94 Performance Dojo

Bottom is the latest process and system monitoring terminal user interface tool, delivering lightweight but beautiful monitoring.

### On the DVD

#### openSUSE Leap 15.5

Leap derives from SUSE Linux Enterprise (SLE) source, conferring stability on users, developers, and sys admins. Along with the EFI/UEFI specification, the new GUID partition table (GPT) schema uses globally unique identifiers to identify devices and partition types. The RPM and repository signing the key has moved from a 2048-bit to a 4096-bit RSA key. A menubar at the bottom of the installation screen lets you specify options to be passed to the installation routines without the need for a detailed understanding of the parameter syntax.



## News for Admins

# Tech News

## CIQ Offers Long-Term Support for Rocky Linux on AWS

CIQ, the company behind the Rocky Linux project, is now providing CIQ long-term support (LTS) for Rocky Linux 8.6, 8.8, and 9.2 images on Amazon Web Services (AWS).

According to the announcement (<https://ciq.com/blog/ciq-offers-lts-for-rocky-linux-8-6-8-8-and-9-2-images-on-aws/>), this LTS “ensures extended life for discontinued major and minor operating system versions, maintaining point release operating system life for at least 2 years.” Customers can subscribe for 24/7 access to the latest images of Rocky Linux point releases through AWS Marketplace ([https://aws.amazon.com/marketplace/pp/prodview-xe2snbtbhdpc?sr=0-1&ref\\_=beagle&applicationId=AWSMPContessa](https://aws.amazon.com/marketplace/pp/prodview-xe2snbtbhdpc?sr=0-1&ref_=beagle&applicationId=AWSMPContessa)).

Additionally, CIQ is now offering enterprise-level support (<https://ciq.com/products/rocky-linux/benefits/enterprise-level-support/>) for Rocky Linux delivered directly from CIQ engineers. This support includes “escalation support, customization, optimization, integration, and professional services.”

See more details at the CIQ website: <https://ciq.com/>.

## Apple’s PQ3 Brings Post-Quantum Security to iMessage

The Apple security team has announced PQ3, a cryptographic security upgrade in iMessage. According to the announcement (<https://security.apple.com/blog/imessage-pq3/>), PQ3 is “a groundbreaking post-quantum cryptographic protocol that advances the state of the art of end-to-end secure messaging.”

Apple says PQ3 is the first messaging protocol to reach what they call Level 3 security, “where post-quantum cryptography is used to secure both the initial key establishment and the ongoing message exchange, with the ability to rapidly and automatically restore the cryptographic security of a conversation even if a given key becomes compromised.”

PQ3 provides “the strongest security properties of any at-scale messaging protocol in the world,” the announcement says. See details at Apple (<https://security.apple.com/blog/imessage-pq3/>).

## Google Open Sources Magika File-Type Detection System

Google has open sourced Magika, its AI-powered file-type identification system.

Magika (<https://google.github.io/magika/>) “leverages the power of deep learning” to help accurately detect binary and textual file types. “Under the hood, Magika employs a custom, highly optimized deep-learning model, enabling precise file identification within milliseconds, even when running on a CPU,” the announcement states (<https://opensource.googleblog.com/2024/02/magika-ai-powered-fast-and-efficient-file-type-identification.html>).

Accurately detecting file types is crucial for determining how to process files, the announcement notes, and tools such as libmagic and the file utility have been the standard for more than 50 years. Magika, however, “outperforms traditional tools with 99% + average precision and recall,” the website says.

The Magika code and model are freely available on GitHub (<https://github.com/google/magika>) under the Apache 2 license. You can try out Magika through the web demo (<https://google.github.io/magika/>) or install it as a Python library and standalone command-line tool using the command: `pip install magika`.



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

**Subscribe free to  
ADMIN Update  
and HPC Update  
[bit.ly/HPC-ADMIN-Update](https://bit.ly/HPC-ADMIN-Update)**



## Microsoft Announces Sudo for Windows

Microsoft has announced Sudo for Windows as part of Windows 11 Insider Preview Build 26052.

According to the announcement post (<https://devblogs.microsoft.com/commandline/introducing-sudo-for-windows/>) by Jordi Adoumie, “Sudo for Windows is a new way for users to run elevated commands directly from an unelevated console session.”

The open source project, which can be found on GitHub (<https://github.com/microsoft/sudo>), will be familiar for users “who want to elevate a command without having to first open a new elevated console.” The company says it has no plans to provide sudo support for Windows Server, however.

For more information on the functionality of Sudo for Windows, check out the documentation: <https://learn.microsoft.com/en-us/windows/sudo/>.

## Linux Foundation Launches Post-Quantum Cryptography Alliance

The Linux Foundation has announced the Post-Quantum Cryptography Alliance (PQCA, <https://pqca.org/>), “an open and collaborative initiative to drive the advancement and adoption of post-quantum cryptography.”

“With the rapid advancements in quantum computing, the need for robust cryptographic solutions that can withstand attacks from future cryptographically-relevant quantum computers has become paramount,” the announcement says (<https://pqca.org/post-quantum-cryptography-alliance-launches-to-advance-post-quantum-cryptography/>).

Launch projects include the Open Quantum Safe project (<https://openquantumsafe.org/>), which supports the transition to quantum-resistant cryptography, and the PQ Code Package (<https://github.com/pq-code-package>), which is focused on building “software implementations of standards-track post-quantum cryptography algorithms.”

Founding members of PQCA include AWS, Cisco, Google, IBM, IntellectEU, Keyfactor, Kudelski IoT, NVIDIA, QuSecure, SandboxAQ, and the University of Waterloo.

## Sys Admins Saw the Biggest Average Salary Increase in 2023, According to Dice

Systems administrators saw the biggest average salary increase (11.2%) in 2023, followed by software developers (6.5%) and program analysts/managers (6.1%).

“Help desk technicians saw nearly 5% average salary growth, as well; to succeed in the role, these specialists must become skilled at solving problems for remote, hybrid (i.e., in the office a few days per week), and full-time office workers,” the report says.

Overall, the top five highest-paying tech occupations, according to the report, are:

- IT management (CEO, CIO, CTO, VP): \$163,526 (avg. salary)
- Solutions architect: \$157,768
- Program analyst/manager: \$148,173
- Principal software engineer: \$145,206
- Cybersecurity engineer/architect: \$140,565
- Sys admins, despite the salary increase mentioned previously, rank #28 on this list with an average salary of \$94,597.

The top five highest-paying skills are:

- Service-oriented architecture: \$137,917 (avg. salary)
- SAP HANA: \$137,626
- PaaS: \$135,868
- Elasticsearch: \$135,541
- Docker: \$135,055

Other highlights:

- 93% of employed tech professionals are either looking for a new job or willing to hear about a new opportunity.
- Salaries for tech professionals working at a technology company are 2.9% higher on average than in other industries.

Read the full report at Dice: <https://techhub.dice.com/dice-2024-tech-salary-report.html>.

## Use of Open Source Software Increased Significantly in 2023

According to the 2024 State of Open Source Report (<https://www.openlogic.com/resources/state-of-open-source-report>), 95 percent of organizations increased or maintained their use of open source software during the past year, and 33 percent said their open source use increased significantly.

This year's report from OpenLogic — in collaboration with the Open Source Initiative (OSI) and The Eclipse Foundation — highlights trends related to the use of open source software, including why organizations choose open source, what challenges they face, and which technologies are most widely used.

The top reasons to use open source, according to the report, are:

- No license cost/overall cost reduction (36.64%)
- Functionality to improve development velocity (30.71%)
- Stable technology with community long-term support (27.64%)
- Access to innovations and latest technologies (26.86%)
- Reduce vendor lock-in (21.29%)

The main areas of open source investment include:

- Databases and data technologies (35.00%)
- Cloud and container technologies (31.64%)
- Programming languages and frameworks (31.50%)
- Operating systems (28.64%)
- DevOps/GitOps/DevSecOps tooling (26.93%)

This year, the survey added a question about open source security tools, with interesting results. According to the report, “27% of respondents selected ‘I don’t know’ when asked which open source security tools they were using in their organization.”

The top open source security tools that were cited by respondents include:

- NMAP (19.35%)
- OWASP Dependency-Track (10.46%)
- pfSense (10.37%)
- Dependabot (10.19%)
- Maven version plugin (9.54%)

Learn more and download the free report from OpenLogic: <https://www.openlogic.com/resources/state-of-open-source-report>.

## Docker Build Cloud Announced

Docker, Inc. has announced Docker Build Cloud (<https://www.docker.com/products/build-cloud/>), which is aimed at reducing time spent waiting on builds to complete.

Docker Build Cloud speeds Docker image builds “up to 39x by offloading workloads to the cloud, regardless of whether developers build locally or through continuous integration (CI),” says the announcement (<https://www.docker.com/press-release/build-cloud-solution-boosts-developer-productivity-accelerating-build-times/>).

Additionally, Docker Build Cloud, which is available now for existing Docker customers, “seamlessly integrates within the existing workflow of developers and incorporates a shared build cache and accelerates multi-architecture builds with native builders.”

## Wi-Fi CERTIFIED 7 Announced

The Wi-Fi Alliance (<https://www.wi-fi.org/>) has announced Wi-Fi CERTIFIED 7, bringing “powerful new features that boost Wi-Fi performance and improve connectivity across a variety of environments.”

According to the announcement (<https://www.globenewswire.com/news-release/2024/01/08/2805409/0/en/Wi-Fi-Alliance-introduces-Wi-Fi-CERTIFIED-7.html>), Wi-Fi 7 technology will deliver “higher data throughputs and support deterministic latency for sophisticated use cases that demand exceptional reliability.”

Advanced features, such as 320 MHz channels and multi-link operation, offer the following benefits:

- 2x higher throughput
- Deterministic latency and increased efficiency
- 20 percent higher transmission rates
- Enhanced spectral efficiency

Learn more from the Wi-Fi Alliance website: <https://www.wi-fi.org/>.



## EU Commissions Nostradamus Project for Quantum Testing

The European Union (EU) has commissioned the Nostradamus consortium to create infrastructure for testing quantum key distribution (QKD), reports Nancy Liu.

“This latest quantum push from the EU aims to pave the way for developing and implementing the European Quantum Communication Infrastructure (EuroQCI, <https://digital-strategy.ec.europa.eu/en/policies/european-quantum-communication-infrastructure-euroqci>), a secure pan-European communication network based on quantum technology,” Liu says.

The EuroQCI, in turn, aims to “reinforce the protection of Europe’s governmental institutions, their data centers, hospitals, energy grids, and more, becoming one of the main pillars of the EU’s Cybersecurity Strategy ([https://ec.europa.eu/commission/presscorner/detail/en/ip\\_20\\_2391](https://ec.europa.eu/commission/presscorner/detail/en/ip_20_2391)) for the coming decades,” according to the European Commission.

Read more at SDxCentral: <https://www.sdxcentral.com/articles/news/eu-invests-200m-in-quantum-technology-to-secure-communications-networks/2024/01/>.

## NIST Identifies Main Types of Adversarial Machine Learning Threats

A new National Institute of Standards and Technology (NIST) publication identifies general types of cyberattacks — so-called “adversarial machine learning” threats — that can be used to attack or manipulate the behavior of AI/ML systems.

The four main types, according to the news statement (<https://www.nist.gov/news-events/news/2024/01/nist-identifies-types-cyberattacks-manipulate-behavior-ai-systems>) are:

- Evasion attacks, which attempt to alter an input after the AI is deployed
- Poisoning attacks, which occur in the training phase through the introduction of corrupted data
- Privacy attacks, which attempt to gain and misuse sensitive information about the AI or the data on which it was trained
- Abuse attacks, which involve malicious insertion of incorrect information into a source

The publication (<https://csrc.nist.gov/pubs/ai/100/2/e2023/final>) “is intended to help AI developers and users get a handle on the types of attacks they might expect along with approaches to mitigate them — with the understanding that there is no silver bullet.”

## GitLab Announces Critical Security Releases

GitLab has announced the release of versions 16.7.2, 16.6.4, and 16.5.6 for GitLab Community Edition (CE) and Enterprise Edition (EE), which contain important security fixes, says Greg Myers in a recent blog post. The vulnerabilities addressed include a critical one that could allow account takeover via password reset without user interactions.

“We strongly recommend that all installations running a version affected by the issues described below are upgraded to the latest version as soon as possible,” the announcement states (<https://about.gitlab.com/releases/2024/01/11/critical-security-release-gitlab-16-7-2-released/>). GitLab.com has already been updated.

The blog post also outlines the following steps to take if you believe your GitLab instance has been compromised:

1. Apply the Critical Security Release to your GitLab instance.
2. Enable two-factor authentication (2FA) for all GitLab accounts.
3. Rotate all secrets stored in GitLab.
4. Follow the steps in GitLab’s incident response guide:  
[https://docs.gitlab.com/ee/security/responding\\_to\\_security\\_incidents.html#suspected-compromised-user-account](https://docs.gitlab.com/ee/security/responding_to_security_incidents.html#suspected-compromised-user-account).

Open source forensics for adaptive detection of threats on CRITIS networks

# Dinos in the Matrix

The open source tool Velociraptor is at the heart of a solution that automatically detects cyber threats in industrial environments, offering a defensive strategy and protecting critical infrastructures. By Michael Mundt and Harald Baier

**Cyberspace is a highly dynamic place:** New attack vectors are constantly coming to light, such as the infiltration of supply chains that back up software products (e.g., the SolarWinds incident) or the theft of a master key for Microsoft cloud services. Critical infrastructures (CRITIS) also need to face up to these threats. Almost inevitably an IT failure will be attributed sooner or later to a cyberattack. For example, the district of Anhalt-Bitterfeld (Germany) was unable to pay out social benefits to 157,000 citizens in 2021 after a cyberattack and had to stop most of its work for two and a half weeks. This incident prompted regulators to intervene and prescribe certifications (e.g., ISO 27001 [1]) and IT baseline protection methods.

## Adapting to Risk

In this article, we look at an adaptive approach that dynamically aligns CRITIS defense with the threat situation by combining information from cyber threat

intelligence (CTI) with methods from adaptive live forensics. In this way, attacks can be detected quickly and security measures initiated at short notice.

For the examples in this article, we use the MITRE ATT&CK knowledge base [2] for CTI and the open source Velociraptor digital forensics and incident response (DFIR) platform [3] in

a lab environment. Velociraptor provides a more detailed and improved view of the status of the system's monitored endpoints. The framework comes with a list of artifacts pre-installed that are configured centrally and executed on the endpoints. Individual queries can also be created with the native Velociraptor Query Language (VQL).

Initial Access 12 techniques	Execution 9 techniques	Persistence 6 techniques	Privilege Escalation 2 techniques	Evasion 6 techniques	Discovery 5 techniques
Drive-By Compromise	Change Operating Mode	Hardcoded Credentials	Exploitation for Privilege Escalation	Change Operating Mode	Network Connection Enumeration
Exploit Public-Facing Application	Command-Line Interface	Modify Program	Hooking	Exploitation for Evasion	Network Sniffing
Exploitation of Remote Services	Execution Through API	Module Firmware		Indicator Removal on Host	Remote System Discovery
External Remote Services	Graphical User Interface	Project File Infection		Masquerading	Remote System Information Discovery
Internet Accessible Device	Hooking	System Firmware		Rootkit	Wireless Sniffing
Remote Services	Modify Controller Tasking	Valid Accounts		Spoof Reporting Message	
Replication Through Removable Media	Native API				
Rogue Master	Scripting				
Spearphishing Attachment	User Execution				
Supply Chain Compromise					
Transient Cyber Asset					
Wireless Compromise					

**Legend**

- Known attack vector of the group
- Functions of the BlackEnergy malicious code
- Functions of the Industroyer2 malicious code
- Functions of the BlackEnergy malicious code
- Functions of the KillDisk malicious code
- Functions of the Industroyer2 malicious code

Figure 1: The Sandworm team's attack vector at a glance.



Various areas of today's networks have critical infrastructures. Operational technologies (OTs) are used in production, for example, where sensors and actuators need to be controlled for automated production. The OTs and control network are carefully isolated from the IT management network by a firewall. Other areas include the security operation center (SOC) network, which provides important monitoring functions. A demilitarized zone secures email communication and enables the integration of external cloud services. In this article, we focus on a typical component of an OT control network: programmable logic controllers (PLCs), which monitor and control industrial processes in production. With the use of a selected advanced persistent threat (APT) from OT as an example, you will discover how to establish effective interactions for detection purposes. You can assume that the selected APT is currently also being used in Ukraine and Western Europe, so the example relates to current and real threats to CRITIS.

## Sandworm Team

As a case study, we look at the Sandworm APT group. This group is widely assumed to be actively

intervening in the war in Ukraine with cyberattacks against critical infrastructure. The Sandworm team is attributed to the Russian military intelligence service and is held responsible for the attacks against the energy supply in Ukraine in 2015/2016 and the global attack with the NotPetya virus. In 2018, this group launched the attack against the Organisation for the Prohibition of Chemical Weapons (OPCW) in The Hague, Netherlands.

The behavior of this group can be studied in detail with the industrial control systems (ICS) matrix of the MITRE ATT&CK knowledge base. In the MITRE ATT&CK nomenclature, *tactics* describe the attacker's objectives (the why of an attack step), whereas *techniques* describe the basic associated procedures (the how). In this case study, we investigate the group's ability to attack operational technology.

The ICS matrix shows attack vectors against OT. The knowledge base lists a number of websites as sources of supplementary information, as well. GitHub offers a suitable browser application, the MITRE ATT&CK Navigator [4], which lets you interactively browse, filter, and highlight data in the ICS matrix and save the results for a good initial insight into the

database. For more in-depth analyses, the data from the MITRE database can be read out with the Python API [5] and analyzed with business intelligence tools.

Figure 1 shows the Sandworm team's attack vector. Known malicious code – KillDisk, Industroyer2, and BlackEnergy – is used to execute some MITRE techniques that are not directly part of the Sandworm group's attack vector, which means these techniques can be used at any time, for example, as a distraction or disruptive measure to accompany the actual attack. The figure highlights potential attack techniques of this group, both in the course of a main attack and as possible flanking attacks.

The Sandworm team's skills are reflected in almost all tactics of the ICS matrix. The group uses several techniques to achieve its goals. In the *Tactics | Initial Access* phase, the group uses the *Spearphishing Attachment* technique. The attackers send customized email with malware integrated in the attachment. If the attachment is opened, the attackers gain access to the system. The group uses social engineering to glean the information needed to word messages in such a suggestive way that the recipient promptly opens the attachment.

## Tactics

In the *Tactics | Execution* phase, the attackers use the *Scripting* technique, which involves injecting scripts into the critical infrastructure and then using existing interpreters to execute the scripts on the target system. Once access to the target system has been established, the script interpreter is exploited to execute malicious scripts during the attack runtime and create new malicious scripts on the target system, if required. In the further course of the attack, the group uses standard logs to obtain detailed information about

Lateral Movement 7 techniques	Collection 11 techniques	Command and Control 3 techniques	Inhibit Response Function 14 techniques	Impair Process Control 5 techniques	Impact 12 techniques
Default Credentials	Adversary-in-the-Middle	Commonly Used Port	Activate Firmware Update Mode	Brute Force I/O	Damage to Property
Exploitation of Remote Services	Automated Collection	Connection Proxy	Alarm Suppression	Modify Parameter	Denial of Control
Hard-Coded Credentials	Data from Information Repositories	Standard Application Layer Protocol	Block Command Message	Module Firmware	Denial of View
Lateral Tool Transfer			Block Reporting Message	Spoof Reporting Message	Loss of Availability
Program Download	Data from Local System		Block Serial COM	Unauthorized Command Message	Loss of Control
Remote Services	Detect Operating Mode		Change Credential		Loss of Productivity and Revenue
Valid Accounts	I/O Image		Data Destruction		Loss of Protection
	Monitor Process State		Denial of Service		Loss of Safety
	Point & Tag Identification		Device Restart/Shutdown		Loss of View
	Program Upload		Manipulate I/O Image		Manipulation of Control
	Screen Capture		Modify Alarm Settings		Manipulation of View
	Wireless Sniffing		Rootkit		Theft of Operational Information
			Service Stop		
			System Firmware		

## Listing 1: YARA Rule

```

Rule BlackEnergy
{
  meta:
    description = "Detects VBS Agent from BlackEnergy Report - file Dropbearrun.vbs"
    author = "Florian Roth"
    reference = "http://feedproxy.google.com/~r/eset/blog/~3/BXJbnGSvEFc/"
    date = "2016-01-03"
    hash = "b90f268b5e7f70af1687d9825c09df15908ad3a6978b328dc88f96143a64af0f"
  strings:
    $s0 = "WshShell.Run \"dropbear.exe -r rsa -d dss -a -p 6789\", 0, false" fullword ascii
    $s1 = "WshShell.CurrentDirectory = \"C:\\WINDOWS\\TEMP\\Dropbear\\\"" fullword ascii
    $s2 = "Set WshShell = CreateObject(\"WScript.Shell\")" fullword ascii /* Goodware String - occurred 1 times */
  condition:
    filesize < 1KB and 2 of them
}

```

the target system (e.g., to perform reconnaissance on the system infrastructure and network architecture) with the *Remote System Information Discovery* technique in the *Discovery* tactic phase. Matching functions are provided by Industroyer2 malware and others.

After reconnaissance of the target system, the group maneuvers through the individual subsystems and expands access to further sub-areas. In the *Tactics | Lateral Movement* phase, the attackers use the *Remote Services* technique to navigate between individual assets and network segments. Services available on the target system – for example, Remote Desktop Protocol (RDP), Server Message Block (SMB) protocol, Secure Shell (SSH) protocol – are used for remote access to data and to transfer the data between segments. The attackers could even manipulate the configurations of the management systems and workstations or download further malicious code to execute on terminal devices in the critical infrastructure. Finally, the group affects the critical infrastructure with the *Loss of View* technique in the *Tactics | Impact* phase, which can result in a persistent or permanent loss of visibility of the status of the ICS system. In other words, the current operating status is effectively concealed, which prevents the operator from intervening and causes damage to the system. The Sandworm group uses various types of malware, including KillDisk,

Industroyer2, and BlackEnergy. KillDisk overwrites memory areas of the operating system with random bytes so that the operating system can no longer be started. Industroyer2 supports the IEC 104 protocol from the supervisory control and data acquisition (SCADA) environment, a control system that records, analyzes, and visualizes data from industrial plants. IEC 104 is the most frequently used communication protocol for periodically sending data, such as status information and events, to the SCADA systems to be monitored. It uses Ethernet and TCP/IP for communication. This protocol can be used to configure criteria for triggering events and basic variables to be reported periodically. The protocol can therefore be used to control SCADA systems remotely.

## Monitoring Remedy

The German Federal Office for Information Security (Bundesamt für

Sicherheit in der Informationstechnik) requires systems used for attack detection to use suitable parameters and be continuously and automatically monitored during operation. The intent is to be able to identify and avoid threats at all times and initiate suitable countermeasures if disruptions occur. By pursuing this objective, users can counteract the identified tactics and techniques of the Sandworm group. To understand how this goal can be achieved, we first test the Velociraptor framework's feature set by individual techniques.

The starting point is the first phase of the attack, *Tactics | Initial Access*. The Sandworm group uses the *T0865 Spearphishing* technique with the help of the *S0089 BlackEnergy* malware. The attackers package the malicious code as an entry in a manipulated Word document, which they send as an email attachment. As soon as the recipient opens the attachment, the malicious code is executed. A YARA rule that detects this process can be found in a GitHub repository [6] (Listing 1). The YARA rule can identify and classify malware.

With the Velociraptor framework, we executed this YARA rule on all connected workstations, which can take place continuously and in parallel. To do this, the Velociraptor server needs to be integrated into the infrastructure and configured for the target environment. Agents are then distributed to the endpoints. Queries can be executed simultaneously on individual or multiple endpoints via the connection to the

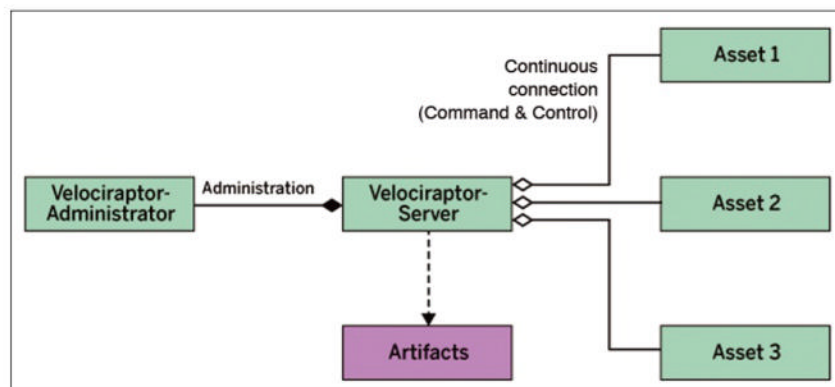


Figure 2: The Velociraptor architecture.



agent. The resulting artifacts are then uploaded to the server and saved there (Figure 2).

The Velociraptor server is managed by an admin web interface. The interface offers functions for browsing the filesystems of and running commands on the endpoints. Moreover, persistent queries are compiled and scheduled at this point.

The VQL scripting language (Listing 2) is used to execute the rule. YARA rules basically use signatures to detect malicious code in filesystems and process environments. If you use a different rule, you can detect other malware in the same way. YARA rules for KillDisk and Industroyer2, among others, are also available.

Finally, the VQL query can be converted into an artifact. To do this, the framework provides a template that you populate with VQL expressions. Preconfigured artifacts are also available. We converted this artifact into a persistent query, also known as a hunt. It can be executed on all connected endpoints. The procedure can be configured and controlled with the management interface.

This first example demonstrates the basic functions for continuous monitoring of a networked system.

## Industrial Controllers

A PLC controls systems and machines in the production environment according to specific algorithms. PLCs go through a continuous process cycle during operation. They are programmed

by the programming languages defined in the IEC 61131 [7] standard: Instruction List (IL), Ladder Diagram (LD), Function Block Diagram (FBD), Sequential Function Chart (SFC), and Structured Text (ST). The IL and ST languages are text-based and the others (LD, FBD, and SFC) are graphical. Functions and function blocks can be used in all languages. Blocks can be written in one of the other languages or delivered as software libraries by the PLC manufacturer. Some manufacturers provide source code; others do not.

Figure 3 shows an example of a program in the graphical LD language. It uses logic-based processing to map the data flow of incoming data to the output data. When the input contact *In1* is closed, the data flows to the

Listing 2: VQL Script to Run YARA Rule

```
LET Yavarule = ''rule Name {...}''
----- File Detection -----
LET Globs = 'C:/Users/ '
SELECT * FROM foreach(row={
  SELECT FullPath FROM glob(globs=Globs)
}, query={
  SELECT str(str=String.Data) As Hit,
  String.Offset As Offset,
  Filename FROM yara(accessor="file", files=FullPath, rules=Yavarule)
})
LIMIT 50
```

output point *Out1*, which then opens and outputs the data. The same process can be expressed in a text-based manner by the ST programming language. Both programs are intuitive and easy to understand.

As you can easily see from this fairly theoretical appraisal, a PLC offers greater flexibility compared with the physical relays previously used. The programming interfaces can be used to design the logical sequences to reflect the tasks in hand. However, this does create potential points of attack. The programming could be changed, or someone could modify the variables or replace software libraries. The attacker could manipulate or delete logfiles, block input or output signals, or launch side-channel

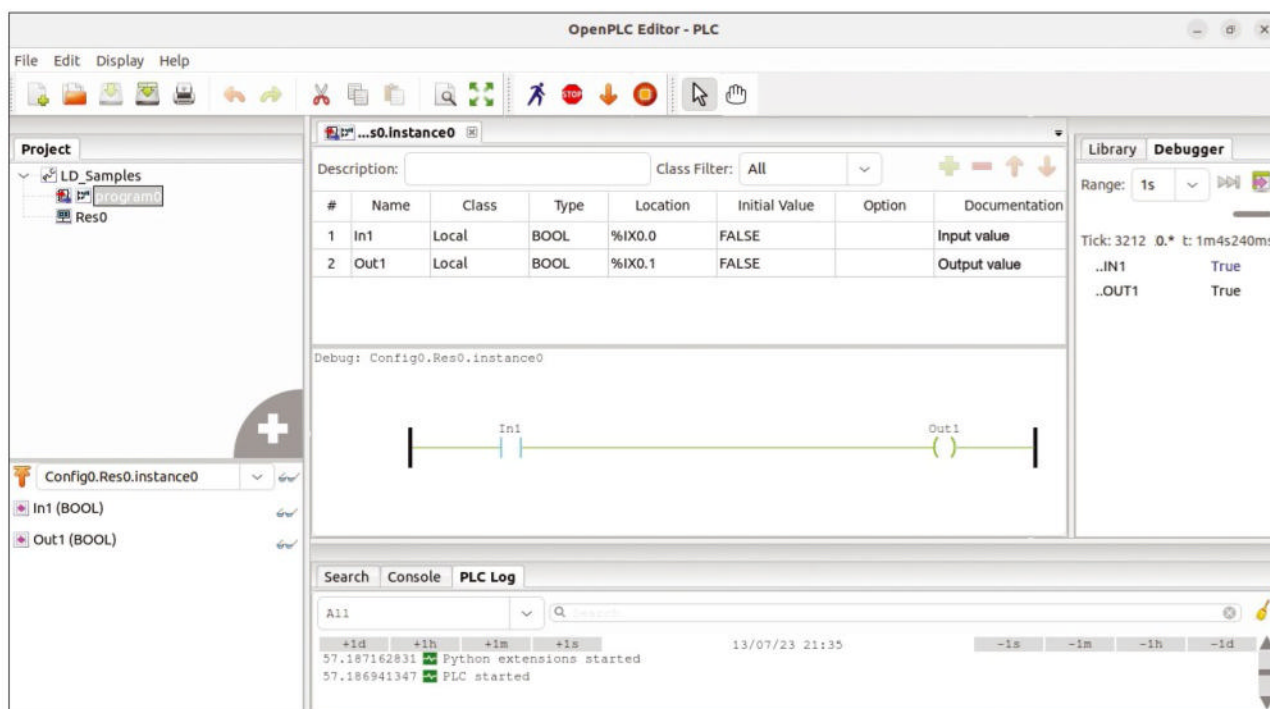


Figure 3: A simple example of a program in the LD language in the OpenPLC editor.

Table 1: Techniques Against PLCs

ID	Technique	Attack Vector	Detection Method
T0803	Block Command Message	Input signal to <i>In1</i>	Monitor the logfile for downtime during operation; monitor the network for error messages when compiling data for the PLC
T0804	Block Reporting Message	Output signal to <i>Out1</i>	Compare logfile entries with measured output data
T0816	Device Restart/Shutdown	PLC runtime environment	Monitor the logfile for termination and restart of the PLC instance
T0857	System Firmware	PLC instance's codebase	Check the integrity of the codebase components

attacks, for example, by interrupting the power supply. In the next step, we examine how the Sandworm group exploits these or other points of attack.

By analyzing data from the MITRE ATT&CK for ICS knowledge base (version 14), we created an overview of all the techniques used by the Sandworm team to target PLCs by selecting all assets that have a *PLC* entry and filtering the *source name* attribute for the *Sandworm Team* group. We

configured this selection in the Python interface with Python pandas data structures. The result was an overview of all the techniques that Sandworm uses against PLCs. The techniques used here relate to four tactics of the attack vector: *Execution*, *Lateral Movement*, *Inhibit Response Function*, and *Impair Process Control*. The four attack techniques – *T0803 (Block Command Message)*, *T0804 (Block Reporting Message)*, *T0816 (Device Restart/Shutdown)*,

and *T0857 (System Firmware)* – are assigned to the *Inhibit Response Function* phase (Table 1). Executing them impairs the PLC's function and blocks its responses.

For each of these techniques, the MITRE ATT&CK framework shows the detection options, including relevant data sources. In the future, it will be possible to call up the connection visually between the technique and the data sources in an even more intuitive way. To this end, MITRE researchers are currently working on a knowledge graph named MITRE D3FEND, which is currently in the beta phase.

We now move to the Velociraptor framework to monitor the designated data sources. The OpenPLC [8] software can be used to set up a test environment to simulate the function of the PLC. The codebase comprises numerous files. For example, the *PLC.so* application is compiled as a shared library so that the application can be loaded into the runtime context of the PLC, called by the *plc\\_main* program, and executed. An MD5 hash is generated during compilation and saved in the *lastbuild-PLC.md5* file. The variables previously configured in the editing environment are located in the *VARIABLES.csv* file. We used the compiler to convert the algorithm to the ST language and saved it in the *plc.st* file.

## VQL Scripts

The first step is implementing a VQL script that monitors, reads (Listing 3), analyzes, and saves the PLC's logfiles. The *upload()* plugin automatically generates the hashes, which can be used at a later date to check whether the saved logfiles have been changed. The Velociraptor framework manages the corresponding metadata for each dataset, which includes the hashes and the name, ensuring proof of file integrity.

In the second step, a VQL script checks the network for blocked connections (Listing 4). These measures are recommended by the MITRE ATT&CK framework for ICS with reference to technique *T0803 Block*

Listing 3: Parsing PLC Logfiles

```
/* - Define the logfile */
LET Logfile <= '/**/PLC/PLC.log'
/* - Read logfile, alert on shutdown if needed */
SELECT * FROM foreach(
  row={
    SELECT * FROM parse_lines(filename=Logfile, accessor='file')
  },
  query={
    SELECT * FROM scope()
    WHERE Line =~ "shutdown initiated" AND log(message='Alarm: PLC shutting down!')
  }
)
/* - upload log messages to server */
SELECT upload(file=Logfile, name='PLC_Logfile.log', accessor='file', mtime=now()) FROM scope()
```

Listing 4: VQL Script for Detecting the Block

```
/* Monitoring ETW provider Microsoft Windows Winsock sockets */
LET DATA = SELECT System, EventData
  FROM watch_etw(guid='{BDE46AEA-2357-51FE-7367-D5296F530BD1}')
  WHERE EventData.ErrorCode =~ "unreachable"
/* Output process using the socket */
SELECT * FROM foreach(
  row=DATA, query=
  {
    SELECT Pid, Ppid, Name, EventData.Protocol, EventData.ErrorCode
    FROM pslist(pid=System.ProcessID)
  }
)
```

*Command Message* with the identifier *DS0029 Network Traffic: Network Traffic Flow*. We implemented the detection measure in the test environment in our specific example. The PLC runs on Windows and uses TCP/IP for communication.

Detection relies on event tracing in the system to monitor the use of sockets. We displayed all processes for which error messages were returned when trying to establish a connection. The actual PLC is blocked and cannot receive any signals. The surrounding components will therefore see error messages when they attempt to communicate with the PLC. We used this fact to support detection. The provider's globally unique identifier (GUID) is revealed on the Windows system with the command:

```
logman query providers
```

The third step checks the integrity of the PLC's codebase (**Listing 5**). Immediately after creating the codebase, we created hashes and saved them. The VQL script we used here computes the hashes of the codebase on request and compares them with the originals. If the two hashes match, integrity is proven; otherwise, there is evidence of manipulation. The script logs the results so that protective measures can be taken immediately in the event of manipulation.

## Conclusions

The three VQL scripts shown here can be used to monitor, analyze and save data sources in the PLC environment. During the entire logging cycle, the results of all detection actions were collected and analyzed in the context of the entire attack vector.

The chosen adaptive approach to attack detection complies with the minimum requirements of the German Federal Office for Information Security. It also ensures significant flexibility in the face of new cyber threats by incorporating the latest findings from cyber threat intelligence (by MITRE ATT&CK in this example). The procedure discussed herein was presented in more detail at the 31st DFN conference, along with numerous other interesting presentations on IT security topics. The paper will be posted online [9]. ■

### Info

- [1] ISO/IEC 27001:2022. Information security, cybersecurity, and privacy protection for information security management systems (ISMSs), [\[https://www.iso.org/obp/ui/#iso:std:iso-iec:27001:ed-3:vi:en\]](https://www.iso.org/obp/ui/#iso:std:iso-iec:27001:ed-3:vi:en)
- [2] MITRE: [\[https://attack.mitre.org\]](https://attack.mitre.org)

- [3] Velociraptor: [\[https://github.com/Velocidex/velociraptor\]](https://github.com/Velocidex/velociraptor)
- [4] Navigator: [\[https://mitre-attack.github.io/attack-navigator/\]](https://mitre-attack.github.io/attack-navigator/)
- [5] MITRE DB API: [\[https://github.com/mitre-attack/mitreattack-python\]](https://github.com/mitre-attack/mitreattack-python)
- [6] Repository for YARA rules: [\[https://github.com/Yara-Rules/rules/blob/master/malware/APT\\_Blackenergy.yar\]](https://github.com/Yara-Rules/rules/blob/master/malware/APT_Blackenergy.yar)
- [7] Standard for PLCs: [\[https://en.wikipedia.org/wiki/IEC\\_61131\]](https://en.wikipedia.org/wiki/IEC_61131)
- [8] OpenPLC: [\[https://autonomylogic.com/\]](https://autonomylogic.com/)
- [9] Mundt, Michael, and Harald Baier. 31st CERT Conference "Sicherheit in vernetzten Systemen" [Security in Networked Systems] (DFN-CERT, Hamburg, Germany, January 30-31, 2024), [\[https://www.dfn-cert.de/informationen/termine/\]](https://www.dfn-cert.de/informationen/termine/) (in German)

### Listing 5: Integrity Check

```
--Generate and compare the ST hash--
/* - Define the file with ST source code */
LET Myfile = '**/PLC/plc.st'
/* - Read the original hash H1 */
LET H1 = '440e20fe034d799af09e029fdab27858'
/* - Compute hash H2 and MD5 hash H3 from it */
LET H2 = SELECT hash(path=FullPath).MD5 AS Hash FROM glob(globs=Datei)
LET H3 = H2[0].Hash
/* - Compare the hash values and log the results */
SELECT * FROM if(condition=(H1=H3),
then={
    SELECT H1,H3 FROM scope()
    WHERE log(message='OK: %v are equal to %v', args=[H1,H3])
},
else={
    SELECT H1,H3 FROM scope()
    WHERE log(message='Error: %v deviate from %v', args=[H1,H3])
})
```





Tools for testing container vulnerability

# Inspector

Vulnerable software and incorrect settings cause problems, but administrators have some tools at hand that can help with container security. By Martin Gerhard Loschwitz

**No matter whose portfolio,** solutions are lining up everywhere to bundle software on customers' systems in Docker or other containers. Where applications installed directly on the system are still typical today, Red Hat Enterprise Linux (RHEL), SUSE Linux Enterprise Server (SLES), and the like will soon be no more than playback tools and comprise only a minimal system kernel and a runtime environment for containers.

One reason these enterprises frequently cite for relying on containers relates to the security benefits the containers claim to offer. At first glance, this assertion cannot be completely dismissed. An attack on an application running in a container causes fewer problems from the administrator's point of view, but planners and administrators would do well not to check the container security box too quickly. Even in this environment, security vulnerabilities remain an issue, especially considering the changes in the typical attack scenario in recent years.

In the past, the goal of an attacker was to gain control over an entire system to exploit it for their own purposes, but today data theft plays

a far greater role. Whether a MySQL instance is running in a container or not, if an external attack succeeds, the data is gone. (See the "Container Complaints" box.)

In this article, I offer a brief market overview of Clair, Anchore, Dagda,

Falco, Harbor, Xray, and Qualys Container Security. All these tools promise more or less the same thing: They take the tedious task of comprehensive security monitoring for the local container zoo off the administrator's plate. Even a brief comparison with

## Container Complaints

Administrators regularly complain about containers because they considerably increase the administrative overhead. A small experiment demonstrates this: If you want to run MySQL in a container, you first need to package it in a container, which immediately changes the rules of the admin game. With a database running directly on the server, you can typically access an SQL shell by typing `mysql -u root`. In a container, this simple approach typically no longer works, or at least not out of the box. Instead, you need port forwarding and a configuration file for the SQL client on the host to enable access with the correct password and username combination. The going gets even tougher where administrators turn to OpenShift, Rancher, and the like. These tools first pack applications into a container and then wrap up the container in a web of bind mounts to integrate configuration files and data directories dynamically where needed. Therefore, the config file for a service will reside in an unintuitive location (e.g., `/var/lib/container-puppet/`

`neutron-api/etc/neutron/neutron.conf`). Incidentally, this monster path comes from a real-world example - to be more precise, from Red Hat's OpenStack distribution, which also comes completely containerized. How is an admin supposed to keep control of applications running in containers with this lack of standardization, while at the same time keeping an eye on potential security issues?

No problem, many manufacturers promise and set out to provide administrators with toolkits for automatically monitoring a container fleet for security problems. They promise no less than an all-in package: The tools offer check individual containers or connect directly to a local continuous integration and continuous delivery (CI/CD) system to check and monitor containers and their images in a fully automated way. Colorful GUIs with great statistics are also included, of course. No matter how you spin it, as an admin, if you are dealing with a setup that comprises almost entirely containers, you definitely need a container toolkit.

Photo by mari lezhava on Unsplash

a few examples, however, shows that not all that glitters in the brochures is gold. Which tools are fit for purpose and which look more like the result of marketing?

## Clair

If you are looking for an old hand in the container security circus, Clair [1] is certainly a candidate (Figure 1). Clair is not an independent software component but part of the Quay project, the container registry that Red Hat acquired along with CoreOS and later placed under a free license. At the time, the people at Red Hat showed some admirable foresight when they made this buying decision. When the purchase was made, many administrators were unfazed by the idea of downloading arbitrary container images off the Internet and running them locally. Naturally, this set off the security manager's alarm bells; after all, running a black box like this in your own environment is a dangerous thing to do.

Quay offered the possibility to set up your own container registries and populate them with your audited and hand-picked images, which implicitly already contributes

to an overall security solution for containers. Clair is the logical extension. It acts as a security scanner for the images located in Quay. The name is derived from the French word *clair*, which means clear, the implication being that if you rely on Clair, you can enjoy a clear view of your containers' security.

Under the hood, Clair comprises a database, a service for indexing images, another for detecting security

vulnerabilities, and a third for sending out notifications about problems found. In typical microarchitecture application style, all of the components can also run in individual containers and scale horizontally. Clair presents no performance problems, even in large environments.

Checking containers is relatively simple: You run a single command that tells Clair to check an image on the spot, for which it then accesses a central, provider-maintained database of security vulnerabilities and compares the software in the image with the version numbers and files specified there. If it finds a match, Clair alerts you to that fact. It also generates a graphical overview of any vulnerabilities found, sorted by priority. For each individual test, a press of the button associated with the test reveals more detailed information on the vulnerability, including the Common Vulnerabilities and Exposures (CVE) number.

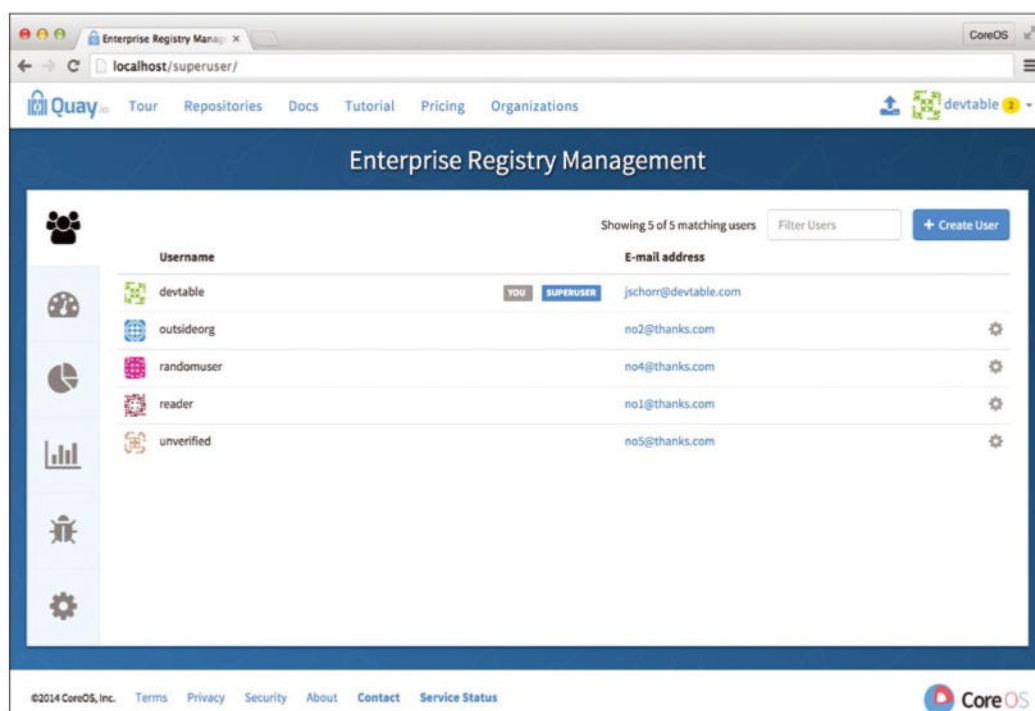
Because of the relatively simple call syntax, Clair can also be integrated easily into CI/CD processes. For example, you can specify that a local build process tells Clair to check each container image generated before

the image is uploaded to the registry. Clair is therefore useful for companies that want to check their container images without having to install a huge framework.

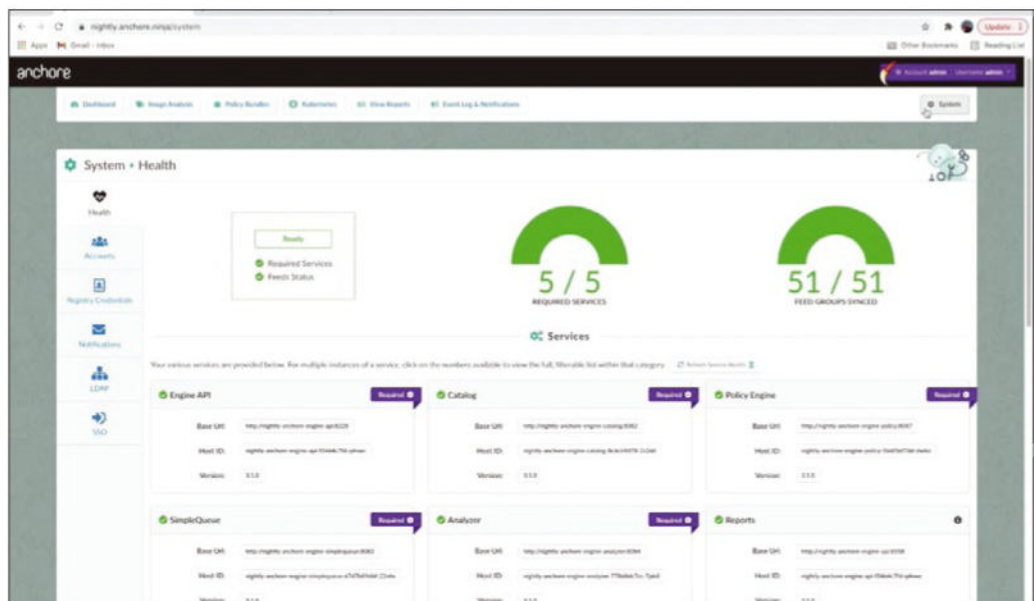
## Anchore

The far more comprehensive Anchore (Figure 2) platform has the Syft and Gripe [2] components under the hood. The manufacturer advertises its product as based on the software bill of materials (SBOM) nested inventory, which is a format that can be used to describe precisely the software and all of its defining details. Anyone who has ever dealt with CI/CD will realize where the journey is heading. As a platform, Anchore loves to dock onto CI/CD environments and integrate itself into the development process, which is where Syft and Gripe collaborate.

In the style of an app based on the microarchitecture approach, Syft first inventories the existing software, creating a complete list of images, the files they contain, and the software versions used. The Gripe security scanner then creates a summary of the threat situation according to data



**Figure 1:** Clair belongs to Quay and came to Red Hat with CoreOS by a roundabout route. The component inspects images for known security problems. © Red Hat



**Figure 2: Anchore offers a comprehensive solution for a security and audit trail in Kubernetes, which, however, requires much infrastructure of its own. © Anchore**

from Syft. Like Clair, it uses provider-maintained threat databases in the background.

However, Gripe has a far larger feature set than Clair. For example, it can reference rules specified by the admin to check source text and files with text content in containers. The rules it applies can, for example, define programming principles. Gripe can also scan source code directories. The underlying idea is clear: If building the container were to produce an image containing a component with security vulnerabilities, you would want to avoid building the image at all.

Instead, you need an error message to draw your attention to the issue during the build process to save unnecessary traffic and load on the build system. The finished image would not be able to be used anyway because of the security problem. In contrast to Clair, Anchore with its Gripe and Syft components offers a way to point out security problems at an early stage in the process, not just when an image becomes available.

The provider refers to this approach as end-to-end security. On the flip side of the coin, though, it means dealing with a high level of complexity as early as at the security scanner level. Whereas Clair can be rolled out relatively quickly and initial checks

can be carried out quickly, admins who use Anchore first have to deal with Anchore's quirks and deployment mechanisms. Ready-made integration in Kubernetes, Docker, and the like is also available for Anchore, but again you need local tools that go far beyond the usual `kubectl`. Although you can't launch Anchore on a whim, the reward for the extra work is far more comprehensive security scans that can take compliance factors into account. The Syft-Gripe combo can also find software that uses undesirable licenses, if so desired.

The solution's big drawback compared with Clair is that, although Syft and Gripe are available under a free license and are easy to install and use, Anchore sells a platform with central control options and the option to visualize any issues found is unfortunately only available as a commercial tool. Therefore, Anchore is more suitable for large environments where you want centralized cover for all security and compliance issues.

## Dagda

Dagda [3], which is more reminiscent of Clair than Anchore in terms of form and scope, is smart but also far more limited in scope. The tool,

written in Python, has a simple modulus operand: You initialize the local vulnerability database and feed in the latest available data, then pass the name of an image, the version to be checked, and the check mode to `dagda.py`. All done. Dagda downloads the image, analyzes the software installed in it, and outputs a list of the vulnerabilities found in the image in JSON format.

However, anyone

hoping that Dagda is limited to local use is very much mistaken. Dagda consists of at least one REST service and one MongoDB database in the background, which the tool references to check the images. Of course, Dagda itself can be operated and rolled out in the form of a Docker container, so at least the initial setup is easy and you can get started far faster than with Anchore, for example.

Quite remarkably, Dagda not only checks images for security vulnerabilities but also detects unwanted software in the form of malware or trojans. Its makers have learned – from the popular attack pattern from the early days of the cloud – about the practice of hiding Bitcoin miners in containers to steal other people's compute power. Dagda puts an end to this ruse, ensuring that images containing undesirable content are not launched. For its tests, Dagda relies in the background on the trojan and virus database from ClamAV, the well-known antimalware toolkit. However, Dagda's update policy is a worry. The last version of the tool dates back to mid 2021; nothing has changed in the Git directory since then. The last release was from the same period. Because Dagda is not backed by a large company, you might worry that the original authors have lost interest in their work.



On the other hand, the software has no unprocessed problem requests piling up in the tool's Git repo. Only a few cases of obviously incorrect use appear, to which there was no response. For the time being, Dagda works well, partly because the databases for the tool's heuristics are maintained by others.

## Falco

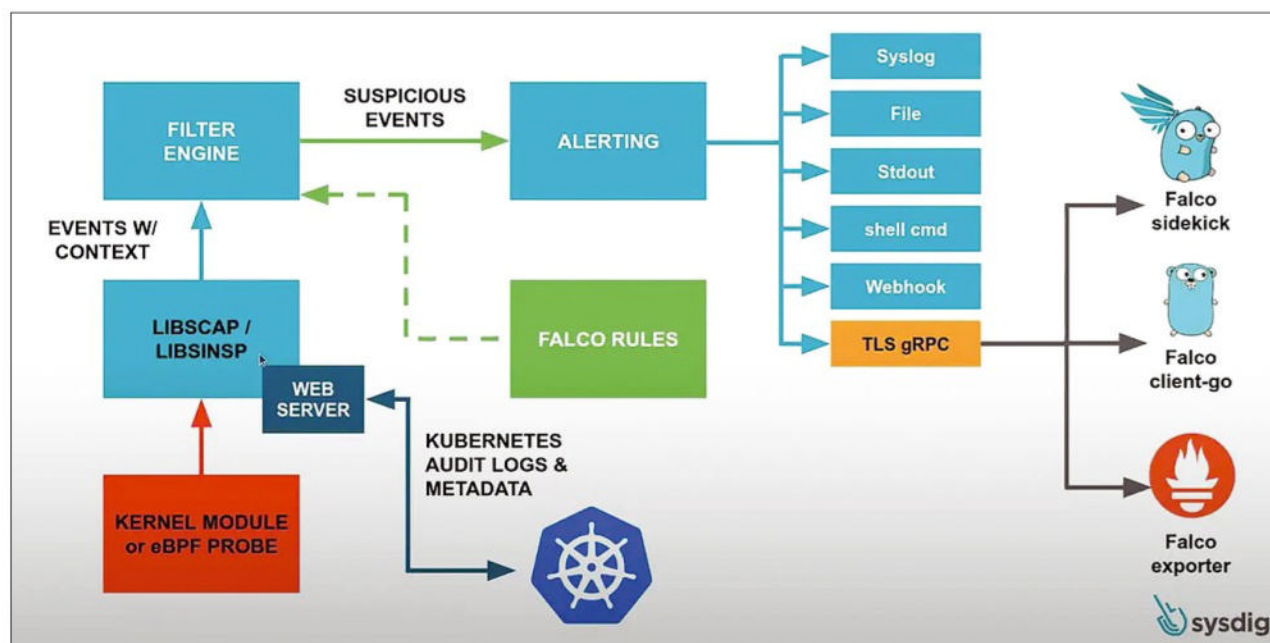
Falco (Figure 3) is also more of a tool in the style of Clair or Dagda and less of a comprehensive platform solution like Anchore. The basic functions are easily explained: The Falco [4] services run in their own containers – for example, within Kubernetes, where they can be rolled out quickly using the Kubernetes package manager, Helm. The package includes a REST API, which is typical today, and an external client for controlling the setup. Under the hood, through, Falco differs considerably from its competitors. It intervenes at the network level and, depending on the configuration, reads the data traffic flowing through, from, and to containers on the basis of the extended Berkeley packet filter (eBPF). Of course, this also means that you have to prepare a platform centrally for the use of Falco, because

Kubernetes does not automatically give you access to the entire data traffic of all containers. Once the tool has been set up, though, nothing stands in the way of security monitoring. Unlike its competitors, Falco does not detect security vulnerabilities in the software of existing containers by comparing them with a CVE database. Instead, the tool detects anomalies in data traffic. The solution also claims to be particularly communicative. From the premise that problem detection is useless if no one sees the resulting notifications, Falco offers connections to more than 30 security information and event management (SIEM) and communication systems. It can therefore be seamlessly integrated into the majority of existing solutions. This arrangement is likely to please both the compliance department and admins, who do not have to set up any special constructs when it comes to Falco's notifications. The icing on the cake is that Falco is available under a free license and can be used free of charge, which makes it ideal as a supplement to a vulnerability scanner such as Clair or Dagda and, to a limited extent, as a complement to a tool such as Anchore, which implements comparable functions differently in some cases.

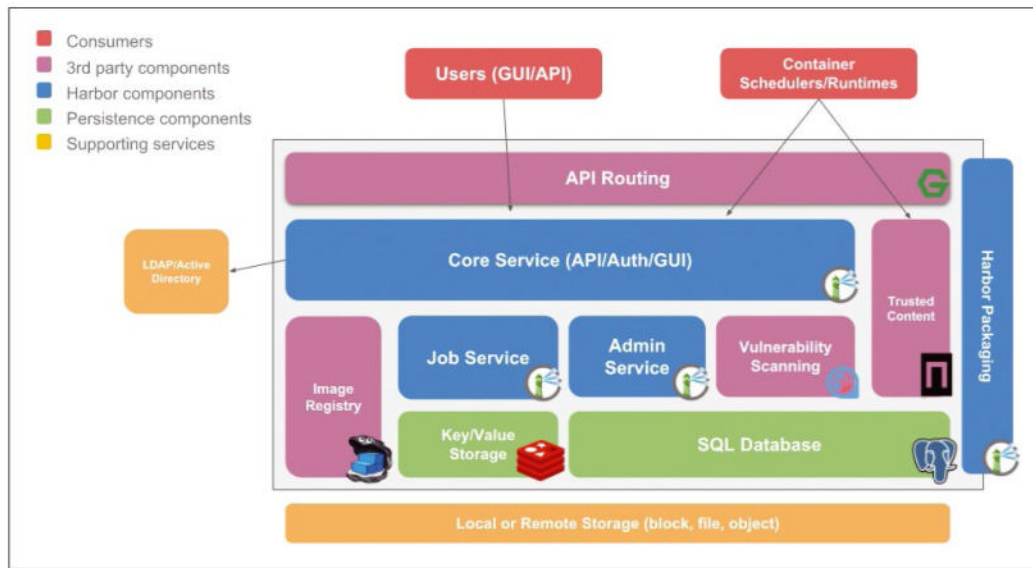
## Harbor

Harbor (Figure 4) approaches the topic of container security from a completely different direction: It acts as a container registry with security checks at the registry level. Harbor [5] primarily checks the images stored in the central registry instead of leaving this task to the admins and developers. As a result, Harbor is more likely to appeal to security and compliance departments in companies looking to provide their employees with a security framework. Harbor also has several options for signing images and validating image signatures. This approach is similar to the direction that Anchore takes with its end-to-end security.

Harbor has various practical functions. The tool is not limited to the images of a single project but is capable of handling multiple clients. Harbor allows you to give different clients in the organization different access credentials for accessing the registry, which ultimately also helps you build an audit trail, because you have records telling you which client downloaded which image from the registry and when. The flipside is a complex installation, because Harbor itself comprises several components



**Figure 3:** Falco does not examine existing containers for images but instead hooks into their network communication, checking it for various problems. © Sysdig



**Figure 4:** Harbor tackles the security issues on its own level and acts as a registry with built-in security features. However, it comes with a mass of separate infrastructure in tow that needs to be operated and maintained. © Harbor

and requires a number of other components in the background, such as a database. Unlike Clair or Dagda, Harbor substantially increases the administrative overhead at the platform management level.

Once rolled out, however, the tool performs well. Under the hood, it relies on Trivy, a tool for detecting vulnerabilities in container images. Trivy works away in the background, creating comprehensive documentation with regard to the work it carries out. Although Harbor is not an obvious choice, what you get is freely usable open source software whose deployment is particularly worthwhile in larger environments.

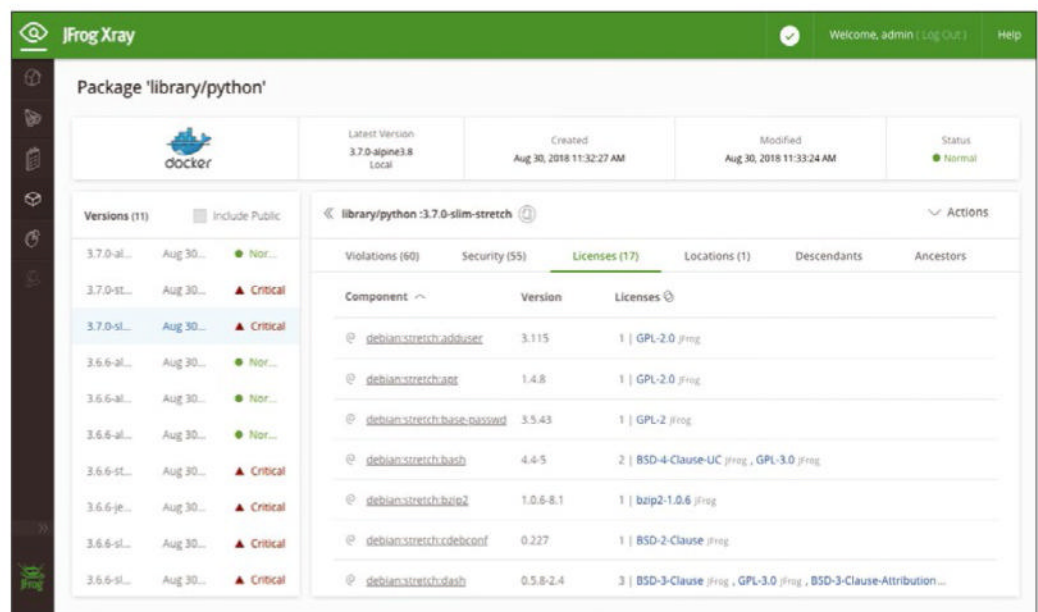
## Qualys and JFrog

The last two test subjects in the test can be dealt with together because they follow very similar approaches and promise users and administrators roughly the same thing. Both Qualys

Container Security [6] and JFrog Xray [7] are platform solutions that can be integrated into and connected to existing containers. The feature set is quite similar: Both products promise to detect vulnerabilities in container images, create curated catalogs of known secure images, and, in some cases, inspect the traffic flowing through the containers. Both products scatter buzzwords as far as the eye can see (e.g., AI- and machine learning-based automatic issue detection); however, the tools do at least offer you a clear-cut GUI

turnkey solution into the enterprise. It is impossible to describe fully the entire scope of services at this point. If you are considering using these solutions, you have no alternative but to have the manufacturer explain and demonstrate the product to you in detail. Even then, the rollout will not be a quick fix, but an extended project. Both JFrog and Qualys want to see cash for their solutions. Of course, everyone has to make a living, but an investment in the context of a company's infrastructure only makes sense if you are going to use the tools and

where you can create reports, check overviews of the issues discovered, and monitor the goings-on in a live view. Qualys Container Security even integrates with TotalCloud, a complete container platform by the same provider. Neither JFrog ([Figure 5](#)) nor Qualys can be integrated into existing environments on the side. Instead, the focus is on the idea of bringing a



**Figure 5: JFrog Xray is a comprehensive platform solution that promises a quick solution to your problems. © JFrog**

their use is centrally mandated. If you are looking for a tool to secure existing platforms, you are probably not in the right place for the holistic approach that JFrog and Qualys take.

## Conclusions

The solutions presented here can be roughly divided into two classes. On the one hand are small tools that can be connected to or integrated into existing environments. The other side is occupied by comprehensive platforms that cover security almost as a side issue.

The combination of Clair or Dagda with Falco seems to make sense, in that both existing vulnerabilities in images and ongoing attacks can be identified. If you prefer a more comprehensive approach, it is best to take a look at Anchore, but you will need to factor in the cost of introducing and maintaining the platform on an

ongoing basis. The same applies to Harbor, which comes with masses of infrastructure of its own. However, as a container registry with end-to-end security and various compliance functions of its own, it stands out from the field of test candidates.

You need to be aware that deploying a single tool is not the way to secure containers and the services in them. Only a smart combination of scanners and monitoring promises success. Some of the tools in this overview are suitable for this scenario. The aim is to secure each individual item of container and service deployment with a tool or a combination of tools.

Fortunately, you have a real choice between comprehensive commercial approaches, in the form of one-size-fits-all packages, and genuinely open source tools from the open source community, which offer operational freedom and flexibility. However,

Dagda, the simplest tool for scanning, might make you worry in terms of its update policy and development activity. You don't need to look far to find a perfect replacement, however, in the form of Clair. ■

### Info

- [1] Clair: [\[https://github.com/quay/clair\]](https://github.com/quay/clair)
- [2] Anchore: [\[https://anchore.com/opensource/\]](https://anchore.com/opensource/)
- [3] Dagda: [\[https://github.com/eliasgranderubio/dagda/\]](https://github.com/eliasgranderubio/dagda/)
- [4] Falco: [\[https://falco.org\]](https://falco.org)
- [5] Harbor: [\[https://goharbor.io\]](https://goharbor.io)
- [6] Qualys Container Security: [\[https://www.qualys.com/apps/container-security/\]](https://www.qualys.com/apps/container-security/)
- [7] JFrog Xray: [\[https://jfrog.com/xray/\]](https://jfrog.com/xray/)

### The Author

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



# Get HPC Update every month!

Tune in to the HPC Update newsletter for news, views, and real-world technical articles on high-performance computing.

**Subscribe free!**



[bit.ly/HPC-ADMIN-Update](https://bit.ly/HPC-ADMIN-Update)

## HPC UPDATE

September 12, 2023

Issue 176



**HPC ILLUMINATIONS  
PAVILION**  
A New Opportunity for  
Underrepresented Groups  
at SC23



**LEARN MORE  
& APPLY**

### This Month's Feature



#### Where Does Job Output Go?


By Jeff Layton

Where does your job data go? The answer is fairly straightforward, but Jeff evolves the question to include a discussion of where data "should" or "could" go.

### News and Resources

- [RIKEN Brings AI to Quantum Error Correction](#)
- [Submissions Are Open for ISC 2024](#)
- [Math Magic with MathLex](#)





## Hardening mail servers, clients, and connections

# Special Delivery

If you don't want to hand over your mail to a corporation, you have to operate your own mail server. Securing it sensibly involves some effort. By Martin Gerhard Loschwitz

**An admin faced with the task of building a new mail server** setup will commonly feel slightly disoriented and even discouraged. However, today, a stable and reliable mail server can be operated on any popular Linux distribution – provided you know what you are doing. In this article, I list the basic considerations that play a role in securing mail servers or that relate to the IT environment.

### Concerns

People around the world have been sending and receiving digital messages for more than 40 years now. And the success of email drew the interest of crooks and thieves. Suddenly, unwelcome advertising messages, fraud attempts, spoofed sender addresses, and many other dirty tricks began to flourish.

Since then, associations such as the Internet Engineering Task Force (IETF), which defines the standards for the Internet, have found themselves trapped in a kind of vicious circle. Time and time again they see themselves forced to extend the standards for email and add patches to iron out design flaws in the protocol with new technology. Simply abolishing the technology and replacing it with something completely new is not an option. A global change-over of this kind would be almost

impossible to organize. Therefore, network users have to live with the disadvantages and various negative effects that come with email.

Some smaller companies outsource the problem to service providers such as Microsoft or Google, who now have huge amounts of experience in dealing with email. Other companies do not want to relinquish control of their data. For better or worse, they have no other option than to operate their own mail servers. Many an admin only realizes in the middle of the process that an ad hoc approach does not work; instead, it requires meticulous planning and perfect technical implementation. This approach applies all the more if the issue of email security is to be taken seriously.

### Conditions

A number of factors need to be considered, starting with the mail server itself. The aim is to secure the service so that attackers cannot hijack and exploit it with just a few simple actions. If you don't want to end up on inspect lists or deny lists in the blink of an eye, you need working validation for your own domain. The way email is transported between the client and server and between the mail servers that forward email messages to each other also needs to be secure. Spam

plays a role, as well, because cyber criminals love to distribute their malware by email, so you have to make sure you have working spam and malware filters.

As anyone involved in the operation of mail servers will be aware, it is difficult to find another task for which so many roads lead to Rome. It's true that you are unlikely to find Sendmail in the wild these days – thank goodness, as many a long-suffering admin would probably add. However, even without the oldtimer, the mail server market is confusing, even if you just focus on Linux and ignore the countless Microsoft Exchange setups. Therefore, you need to define a few requirements, particularly with regard to the software to be used.

Postfix is now widely used and is cutting edge in terms of both features and security. The service can be operated on any recent enterprise distribution – I use Ubuntu 22.04, but you can apply all the advice to other distributions, too. ClamAV is used in conjunction with SpamAssassin as a tool for filtering mail. Where action relating to name servers is required, I will not be providing specific commands or instructions and would instead advise you to contact the domain name system (DNS) administrators who are usually exclusively responsible for this task.

Photo by Jesse Ramirez on Unsplash

## Beginnings

I start with a freshly installed Ubuntu 22.04 on which to install a mail server. Before you tackle the mail server itself, the first step is to harden the system. The usual rules apply: Restrict access to the system to as few people as possible. If your company already has a central user management system, it is highly recommended that you connect the mail server to it, so you can enable and disable accounts centrally.

A mail server without direct access to the Internet cannot usually be operated without massive changes to the firewall. However, if the server does need to be accessible from outside your environment, you will at least want to restrict external access to ports that are not relevant for mail. In particular, you will want to configure the firewall or the host itself to deny access to port 22 (i.e., the SSH port). Simple Mail Transfer Protocol (SMTP) ports 25 and 587 should be open. If you also intend to operate with Internet Message Access Protocol (IMAP) on the system, you need to allow access to ports 143 and 993. The same basic rules apply to security measures on mail servers as for all other servers. If you use Ubuntu, for example, make sure you configure AppArmor correctly instead of disabling it across the board. The same applies to SELinux on Red Hat Enterprise Linux (RHEL). Next on the agenda are the security settings of the mail server itself. Operating the mail server without user administration – as with the classic mail protocol – is virtually out of the question. Instead, you will want the ability to send email conditional on successful authentication on the client side. The problem has long since been solved in the Linux world but is still a long way from being used everywhere. More specifically, this requires Simple Authentication and Security Layer (SASL), a bridge construct that acts as an intermediary between the mail server on the one hand and a variety of possible authentication back ends on the other.

As a rule, companies today manage their customer data in some form of directory service – usually by Active Directory or with the Lightweight Directory Access Protocol (LDAP). An SASL plugin for LDAP connects the mail server to the user directory. SASL's LDAP back end can also be linked to Active Directory through its LDAP compatibility interface. The result of all this hard work is that users can only send email if they have an active and valid account in the user directory. Incidentally, an alternative approach to this procedure if you use a pluggable authentication module (PAM) to connect your system to LDAP lets you simply connect SASL to PAM and indirectly inherit the LDAP users.

## Jailing Services

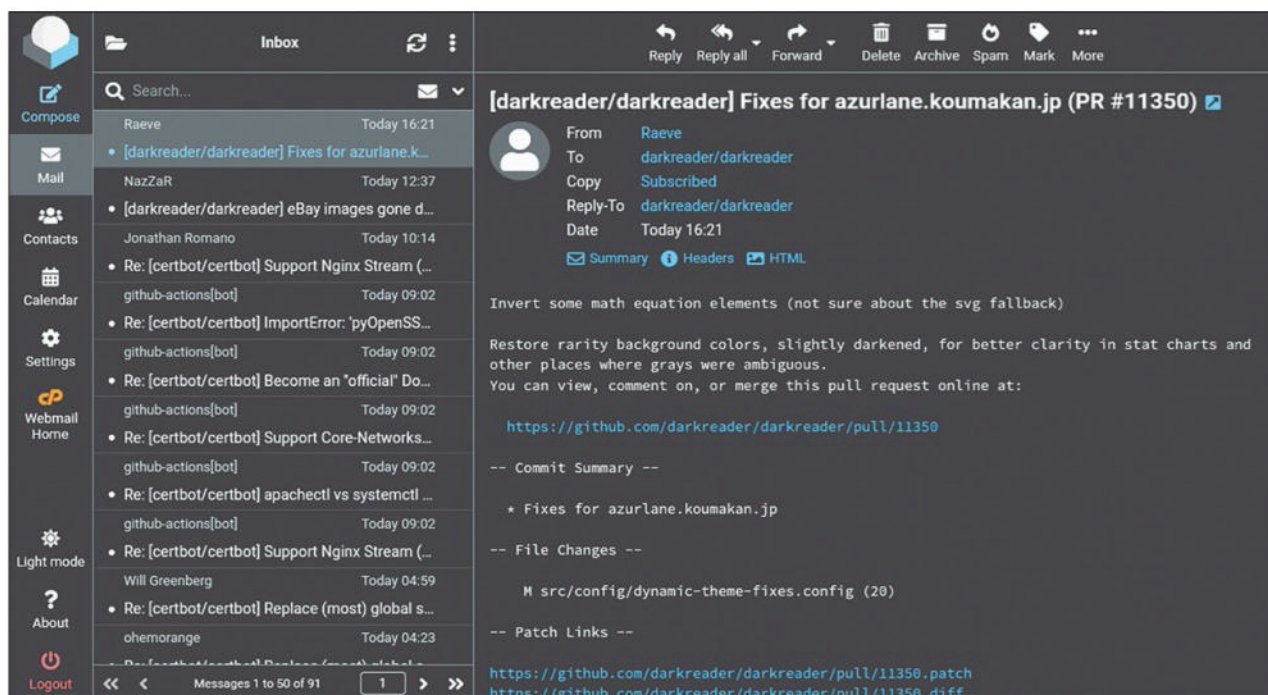
SASL and LDAP make configuring Postfix a tad more complex. Most Linux distributions, including Ubuntu, tend to lock Postfix in its own chroot environment, which is a very superficial form of virtualization. A process locked in a chroot jail views the chroot folder as the root folder of its own filesystem.

This arrangement basically has a similar effect to using a container, even if a chroot does not provide its own namespaces for process IDs or network interfaces, unlike real containers. Nevertheless, the use of chroot for Postfix offers additional security. If chroot is active, a successful attack on the mail server keeps the attacker from gaining immediate access to the host's filesystem. The attacker simply can't see the rest of the filesystem because of the chroot.

Of course, the use of chroot means that all the files to be accessed by Postfix also need to reside within the chroot environment, which in the standard configuration affects the socket that the SASL daemon uses to communicate with its clients. Fortunately, this problem has plenty of slot-in solutions. For example, SASL can be configured so that it stores its socket in the Postfix chroot directory, which is also the recommended configuration.

Speaking of configuration: You can make your life far easier by automating the process of configuring the mail server. Ansible can be put into operation quickly for individual systems; you can find plenty of ready-made Ansible roles online for Postfix on Ubuntu. If you rely on Ansible or some other automation tool, you get all the advantages of automation at virtually no cost and mitigate the risk of careless errors. This approach even gives you a contingency plan: If a system falls victim to an attack and is considered compromised, a replacement can be set up quickly with an existing automation system.

Incidentally, similar considerations apply if the same machine will also be acting as an IMAP server. Dovecot is recommended as a standard product, and it offers security options comparable to running in a chroot jail. If you want to handle the clients on the same system (e.g., with a web-mail client), you can use tools such as Roundcube ([Figure 1](#)). Ideally, however, you will want to make sure the basic details of secure system administration are taken into account. If you are not worried about working with containers, you now have another option for hardening Dovecot and Postfix: running mail services in containers – typically, Docker containers on Ubuntu ([Figure 2](#)). The required directories (e.g., the data directory for Dovecot, which the service uses to access mail messages) are integrated into the construct as bind mounts. Dovecot and Postfix are then completely isolated from the system, which also makes features such as high availability easier. If in doubt, you can move the container in question from one host to another, taking the IP address with it. The admittedly somewhat unloved Pacemaker cluster manager supports this ability in a relatively uncomplicated way. The ideal solution is to combine containerization and automation. Automatically rolled out containers, which Ansible or an alternative product will also configure appropriately, come close to the ideal of an indestructible infrastructure and help you avoid sleepless nights.



**Figure 1:** If you want to build an email server-in-a-box, Roundcube is a good and secure method of connecting a web-based client. © Roundcube

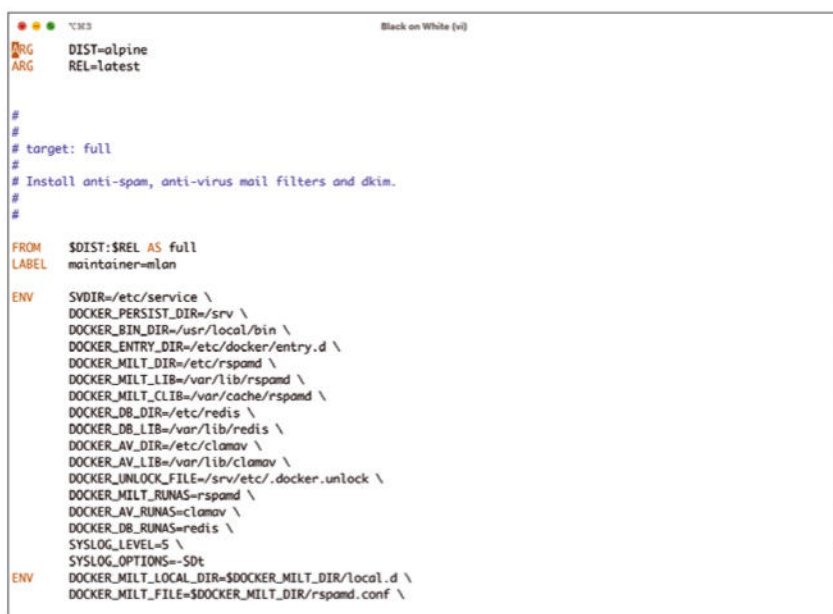
To supplement this article, countless how-tos can be found on the Internet that will give you a good overview of the most useful configuration directives for Postfix, including, for example, setting Postfix up to reject clients that do not say EHLO before sending email. These attempts are either poorly programmed clients or spammers. Lots of little things like this can be used to good advantage every day, but even experienced mail admins don't know them all.

## Securing the Medium

Administrators who automatically roll out Postfix and Dovecot in containers while providing the solutions with working configurations have already achieved something good, but it is still too early to put your feet up. With mail servers, the focus is not only on securing the service itself, but also on securing the ability to send and receive messages. Bad habits relating to email mean that, besides needing the mail server, you also need a suitable DNS configuration, defense against spam, and secure connections to the clients and other mail servers.

Much like web servers, it is not advisable for today's mail servers to talk to any other party in plain text. Anyone who has access to the network connection between the servers could simply sniff any messages transmitted in the clear. It makes more sense to run the mail server with the Secure Sockets Layer (SSL) protocol in place, or at least to install an SSL terminator

upstream of it. The best way to do this depends on your setup. If you are already operating a containerized mail server, you can use a mesh service such as Istio for this task. It terminates SSL connections and takes care of the issue completely independently so that Postfix itself does not even need an SSL configuration. If Postfix is running as a service



**Figure 2:** If you run your mail services in a container (as shown here with Rspamd and Postfix), you add an additional layer of security to the system that makes it more difficult for intruders to gain access.



directly on the system, it can be made SSL-capable with just a few steps. However, you then need to disable the receiving and sending of email over unencrypted channels. The same applies to a third conceivable scenario in which a load balancer upstream of the mail server takes care of SSL. However, be careful: If you want to use DNS-based authentication of named entities (DANE), the mail server must support STARTTLS (a protocol command that switches from plain text to a Transport Layer Security (TLS) or SSL connection), in which case you will need to provide an SSL certificate on the mail server. With STARTTLS, opening the connection is an unencrypted process, but all communication after the STARTTLS command is encrypted.

## Receiving and Sending

When sending, it is important to demonstrate the authenticity of a sent email to the mail server at the other end. As described earlier, email in its original form did not provide for any measures that would have forced the sender of an email to use a specific sender address, opening the door to attacks and ultimately leading to the emergence of several semi-official techniques, some of which are now considered official Internet standards. One of the simplest methods is the Sender Policy Framework (SPF; [Figure 3](#)), which should not be missing from any serious email setup. Put simply, SPF uses a DNS TXT record to store an entry for the sender domain the servers are allowed to use when sending email. Therefore, you need to set up your own DNS server so that only your mail server appears in the SPF entry for your domain. Conversely, your Postfix needs to check the SPF entries of the servers at the other end and reject email if it comes from an address other than the one in the SPF entry.

DomainKeys Identified Mail (DKIM) does something very similar. You store the public part of a cryptographic key as another TXT entry in the domain. When email is sent,

the server computes the hash of the email content and inserts the digital signature into the header fields of the email before it is sent. The receiving mail server can then check the content of the mail on the basis of the public key of the sending mail server and the digital signature. If the signature and key match, DKIM assumes that the sending server is authentic and accepts the mail; otherwise, it rejects the message. Like SPF, DKIM has become a common tool in everyday mail traffic that prevents misuse and should therefore not be missing from any mail server configuration.

## DMARC, DANE, and More

The third technology in the group is somewhat more controversial: The Domain-based Message Authentication, Reporting and Conformance (DMARC) protocol, which also involves storing a TXT entry on your domain's DNS server. However, the approach is far more complex than that of SPF and DKIM; in fact, DMARC even incorporates both technologies. In addition to the matching modes for DKIM and SPF, a DMARC entry for a domain contains specific, standardized instructions (e.g., regarding email forwarding). If these rules are violated, a DMARC-enabled mail server drops the message. This outcome occasionally leads to problems in everyday life, because DMARC can cause legitimate mail to fail. Mailing lists, for example, forward email from the sender to many

recipients. Anyone who has ever received email from a mailing list in which the sender is the mailing list itself, with *User via ...* in the *From* line instead of the original author now knows the reason. These constructs are needed for DMARC to work. Because of the catastrophic flood of spam, however, DMARC has now also become widely established, and new mail servers should use it if at all possible.

Finally, DANE stands slightly apart from the other three solutions and has nothing to do with authenticating the sender; rather, it enforces the use of an SSL-encrypted connection from the sender's point of view. To this end, it makes use of DNSSEC, the standard intended to secure the authenticity of DNS entries through digital signatures, and attempts to solve a problem similar to that DKIM tackles. DNS is an ancient protocol; its original form had virtually no protection against misuse. DNSSEC corrects this problem by using DANE to ensure that a remote mail server can only be considered for delivery of mail to a specific domain if its signed DNS entry explicitly allows an SSL-encrypted connection to be opened by STARTTLS. If the domain does not support it, the outgoing mail server immediately cancels sending. This arrangement prevents man-in-the-middle attacks and can even be a building block for making your setup more compliant with the European Union's General Data Protection Regulation (GDPR). It should therefore be a standard tool on new

<code>v=spf1 ip4:192.168.0.0/16 include:_spf.google.com ~all</code>
<code>v=spf1 ip4:192.168.0.0/16 include:_spf.google.com include:sendyourmail.com ~all</code>
<code>v=spf1 a:mail.solarmora.com ip4:192.72.10.10 include:_spf.google.com ~all</code>
<code>v=spf1 include:servers.mail.net include:_spf.google.com ~all</code>

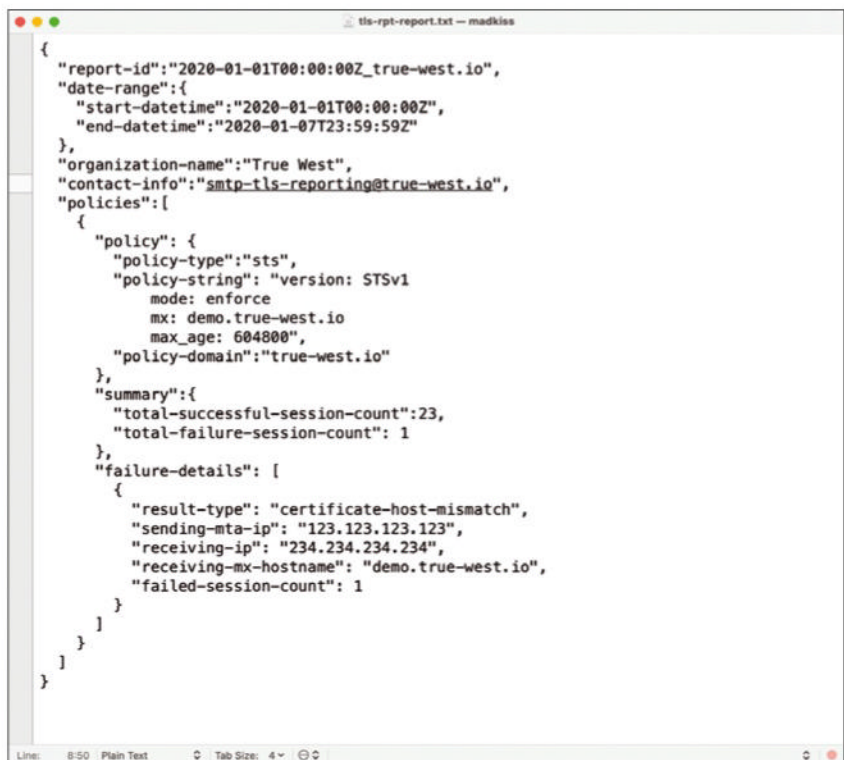
**Figure 3:** Industry leader Google is leading the way: SPF entries are a good way of preventing the misuse of sender domains.

mail servers. In return, you have to accept the fact that mail might not be delivered because of a lack of DANE support on the receiving end. Finally, TLS reporting (TLS-RPT), which acts similarly to DANE, is now considered an accepted standard and relies on state-of-the-art technology. Put simply, with TLS-RPT in place, an outgoing mail server checks whether the message transfer agent (MTA) on the other side supports the STARTTLS command. If so, it uses TLS to encrypt the message and sends it to the other end, where it is unpacked and delivered. By doing so, TLS-RPT enforces encryption of the content to be transmitted. In particular, it rejects clients frequently found on the network that try a hack to downgrade communication from the encrypted variant to plain text while establishing a connection. This hack is still regularly used today, particularly in cases of attempted fraud.

Reporting also plays an important role in TLS-RPT. When TLS-RPT is active, the server admin receives regular reports of incidents (Figure 4), such as when a client has attempted to intervene, as described above. This standard helps detect malfunctions and missed email at an early stage and, where needed, helps the admin know when to change the configuration.

## Protecting the Receiver

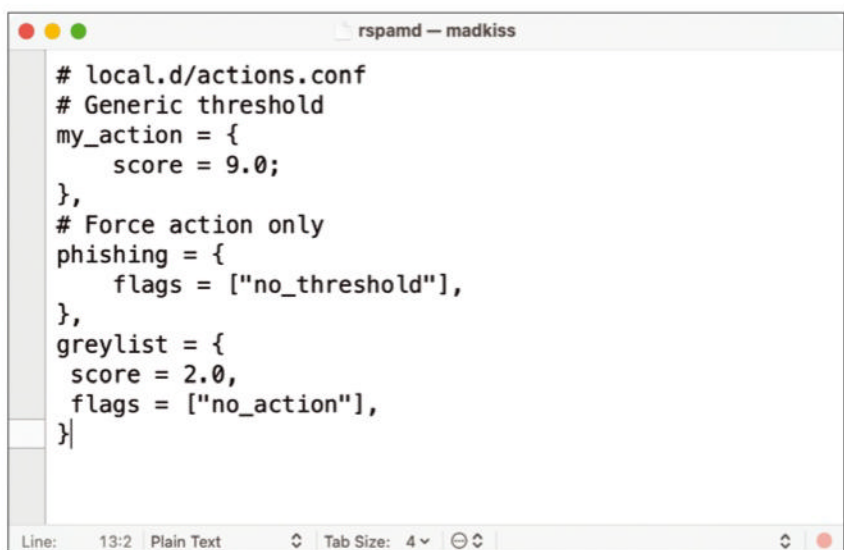
Securing and verifying receiving and sending addresses does not change the fact that billions of malicious email messages find their way into the inboxes of potential victims every day. At the end of the day, doing something about this situation is one of the mail server administrator's obligations. Conveniently, you can draw on tried and tested solutions that have proven their value over many years. Most admins have probably heard of SpamAssassin, a tool typically used in conjunction with ClamAV, a virus scanner. Just one more component is missing for happy mail delivery: the Amavis daemon, Amavisd, which acts as a bridge between Postfix, SpamAssassin, and other components that play a role in the mail delivery process.



**Figure 4:** TLS-RPT generates reports on failed email deliveries, ensuring that admins are immediately notified of potential configuration issues.

The whole setup is basically simple: Postfix receives an incoming email and forwards it to Amavisd, which then calls SpamAssassin, ClamAV, and other components and lets them check the entire content of the email. If one of the called services finds malware, or SpamAssassin identifies unwanted content, it marks the mail header accordingly before Amavisd returns the

message to Postfix along with all the new headers. The final decision on what happens to the mail then rests on Postfix and its configuration. SpamAssassin grades email according to various factors that confirm or allay the suspicion of spam. As the admin, you determine the limits by which a local setup classifies email as definitely spam, suspected spam, or



**Figure 5:** Modern antispam tools such as Rspamd only work efficiently and well if you train them properly to help them evaluate messages with realistic assumptions.

probably okay, but it is important to train a setup of this type yourself and ideally also have it trained by your users. Solutions such as SpamAssassin are based on Bayesian filters that cannot work properly without appropriate training. In an environment like this, it makes sense – from the user's point of view – to move email that has been incorrectly identified as spam to the correct inbox and to configure the filter so that it uses the contents of the spam folder for its own regular training (Figure 5).

An alternative solution based on Rspamd (a compact replacement for Amavisd), SpamAssassin, ClamAV, and other components is far more modern. Rspamd is installed as a mail filter in Postfix and has a strictly modular structure, with modules adding features such as an antivirus option and a filter for potential spam messages. In principle, Rspamd works similarly to SpamAssassin. Unlike

Amavisd, it also has a local delivery agent (LDA) mode. It does not dock onto the Postfix mail server, but to the delivery service – usually Dovecot as the IMAP server. The disadvantage of this solution is that email first checked in the message delivery agent is already considered accepted and can no longer be rejected absolutely. Where this is not a problem, Rspamd offers a modern and powerful alternative to SpamAssassin. If you want to containerize your mail server as described earlier, you need to make sure it also runs the additional services in the container and establishes a communication path to them. Again, some established automation providers offer ready-made solutions for this scenario.

## Conclusions

If you want to make sure your mail server works well and is secure, you

can look forward to a fair amount of work. The patchwork of extensions to the original email standard contributes significantly to the confusion and makes operation more complex, but there's no point in complaining about the inevitable. If you have to run a mail server, you have no choice but to jump through several burning hoops. Thanks to the wide choice of ready-made automation solutions and modern technologies such as containers [1], you can now get started far faster than ever before. ■

## Info

[1] A container with Postfix and Rspamd:  
[<https://github.com/mlan/docker-rspamd>]

## The Author

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



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?

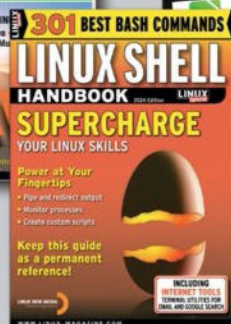
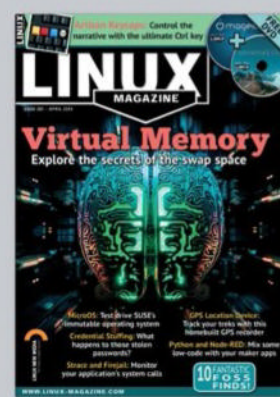
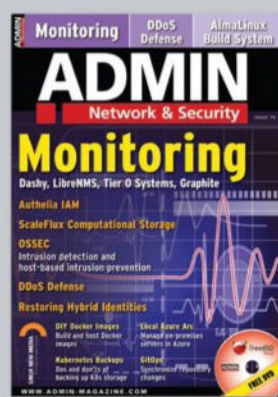
➤ [shop.linuxnewmedia.com](http://shop.linuxnewmedia.com)



DIGITAL & PRINT  
SUBSCRIPTIONS

SPECIAL EDITIONS

➔ [shop.linuxnewmedia.com](http://shop.linuxnewmedia.com)





Centralized monitoring  
and intrusion detection

# Alarm System

Security Onion bundles numerous individual Linux tools that help you monitor networks or fend off attacks to create a standardized platform for securing IT environments. By Erik Bärwaldt

**Ensuring the security** of a company's IT infrastructure becomes increasingly important as new threats and forms of attack continually emerge. Linux already has numerous tools for detecting anomalies in networks across platforms that evaluate logfiles, among other tasks. However, many of the security tools that make life easier for administrators are scattered across the Internet and not easy to find. The Security Onion [1] project addressed this shortcoming back in 2008. Originally based on Ubuntu, the security suite, which now runs in Docker environments, bundles professional tools for monitoring the IT infrastructure, including logfile analysis and intrusion detection [2]. The various ways of using the system include cloud images, which are intended for use in the Amazon, Google, and Azure clouds, and a downloadable ISO image that you can install on a dedicated host and deploy in virtual environments such as VirtualBox or VMware.

## Strategies

When collecting, aggregating, and analyzing data, Security Onion can

take both a host-based and network-based approach. The suite uses three tools for host-based intrusion detection: (1) Wazuh [3] is a fork of the OSSEC [4] intrusion detection system; it monitors hosts and sends data to a server in the event of anomalies. A cross-platform agent is installed on the computers for this purpose. By analyzing the log data, Security Onion detects malware and identifies further vulnerabilities that need to be addressed. (2) Osquery [5] is another host-based tool that queries and logs the system status. (3) Beats [6] uses Winlogbeat to monitor Windows-specific logs and files. Filebeat, on the other hand, is used across all platforms. Both tools transfer their data to the Logstash server integrated into Security Onion. Like other tools in the suite, Beats uses the Elasticsearch search and analysis engine. Security Onion comes with five tools for network-based monitoring: (1) OpenCanary [7] is a honeypot for intrusion detection; (2) Stenographer [8], developed by Google, focuses on collecting large volumes of data; (3) Strelka [9] scans content and prepares it for detailed analysis; (4) Zeek [10] is dedicated

to monitoring and analyzing large volumes of data and can limit the data volume with its own scripting language, which allows you to generate customized logfiles; and (5) Suricata [11] integrates an intrusion detection/intrusion prevention system and uses signatures to find anomalies in network communication.

## Matter of Opinion

Some of the individual tools in Security Onion work with specific web interfaces that are intended to provide their own overview. The security suite, on the other hand, offers a standardized web interface, the Security Onion Console, that not only simplifies data analysis, irrespective of the tools you use, but also supports manual searches for vulnerabilities and anomalies and offers alerting functions. The Security Onion web interface uses standardized tools to display content, primarily relying on Kibana [12] and Grafana [13]. Kibana displays the incoming data on various dashboards, and Grafana is responsible for analyzing the statuses of the system and monitors their performance.

Photo by Stocchi Lam on Unsplash

## Automated

Security Onion lets you define detection patterns for vulnerabilities and develop solution strategies for eliminating specified problems with the help of a playbook. The playbooks comprise several plays that handle different tasks, so the security suite does the work for you. Security Onion comes with around 600 plays out of the box; you can view the results at any time on the dashboard of the web-based interface.

## Installation

The ISO image [14] (version 2.4.30, weighing north of 11GB, was current at press time) can be launched from a removable medium after downloading and checking the signature file for verification.

You must comply with a number of specifications [15] for the target hardware. Production use of the security suite requires at least four CPU cores. The minimum RAM in most application scenarios is 12 to 16GB; ideally 24GB of RAM or more should be installed. The project specifies a mass storage capacity of at least 200GB. Some application scenarios also require two network connections. Security Onion only supports locally installed mass storage devices as installation media. According to the installation instructions, machines with distributed storage capacity, such as NFS drives, are not suitable because of potential performance issues and the complexity of the expected configuration. This problem also rules out the use of RAID controllers. In our lab, the installation routine stopped working shortly after launching on a computer with two physical drives combined in a RAID array.

Once all the hardware requirements are met, you launch the target system from the prepared removable medium. A GRUB boot menu offers to install the system permanently; you can choose between a basic graphical version and a desktop version. If required, you can also integrate the

desktop version into a system that already has the Security Onion Console in place.

For newcomers, I recommend the automatic install; you can enable it by selecting the first entry in the boot menu. The system setup is largely automatic; you only need to enter a username and an admin password. After completing the basic installation, reboot the computer and log in at the prompt with the credentials of the newly created admin account. A setup wizard designed as an ncurses application then appears (Figure 1).

The wizard can only be controlled by the keyboard to add numerous additional components and a Docker environment to the system. To begin, select the *Install* option in the second dialog window and the *Import* option in the third. In the next step, confirm that you have access to the Internet by selecting the *Standard* option. Please note that Security Onion only identifies wired network connections during the install; it cannot be used with WiFi access.

In the next two steps, first confirm the license terms, enter the hostname, and select the network interface. If the system has two or more LAN connections, select the first one that displays the *Link UP* status. You can enter a static IP address, specify the gateway, and define further access

data for this network interface in the next few windows.

To complete the configuration, the wizard displays a brief summary of the settings and finishes setting up the system after you confirm your entries. At the same time, the wizard runs a system update. The entire installation can take several hours, depending on the available computing power and speed of your Internet connection.

## Use

After installation, you can connect to the Security Onion host from any workstation on the LAN by entering the IP address or hostname in the web browser, as specified during configuration in the setup wizard. Authentication credentials are the email address defined with the setup wizard and the matching password. You are then taken to a very clear-cut admin interface (Figure 2).

Top left in the browser window you will find elements that apply to the entire tool collection, such as the *Alerts* display, the *Dashboards*, and the settings dialog, where you can create and manage user accounts. You can also configure your network nodes (*Grid*) or store license keys for external packages. Use the *Downloads* option to set up Elastic agents on external hosts for monitoring. Links to download the

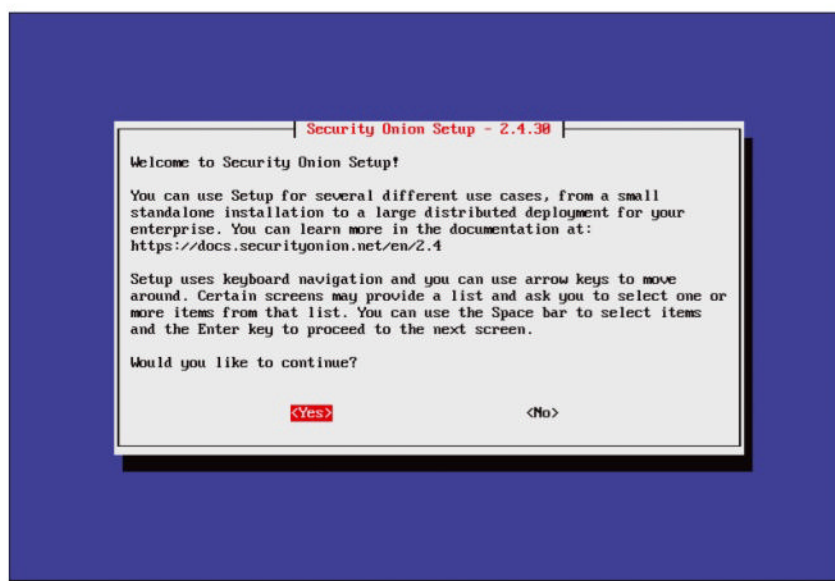


Figure 1: After the basic installation, you can complete the setup with the help of a wizard.

agents for all of the popular operating systems then appear in the right-hand part of the window.

The lower section takes you to the individual tools, some of which have their own web interfaces. Please note that before you can use some of the tools, you will need to install the matching agents up front.

## Overview

The *Dashboards* group provides an overview of the installation. Graphical displays of the respective databases appear on the right-hand side of the browser window when you enter a category and define time intervals for the individual data categories at

the top of the window. The security suite then displays the aggregated data and an analysis lower down in the window. Numerous categories are enabled by default to give you a quick overview of the different view formats. The filtering options in the individual views then extract specific data (Figure 3).

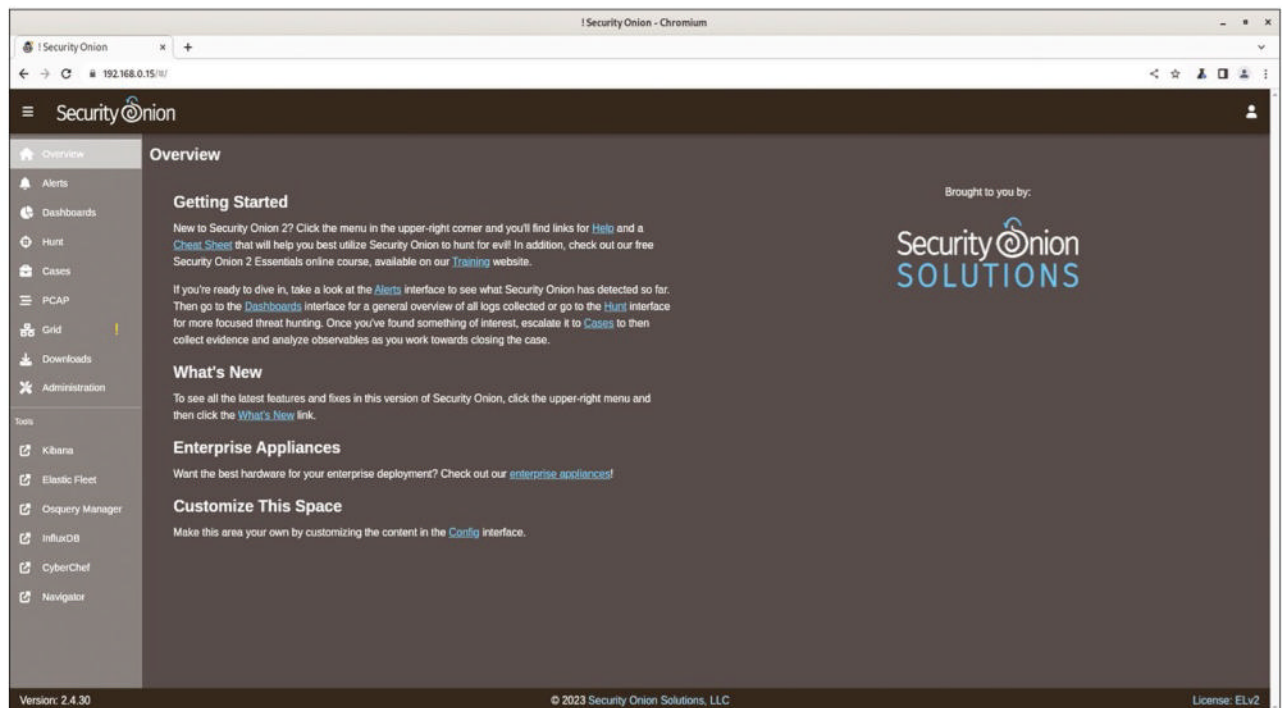


Figure 2: Managing the system in the straightforward admin interface.

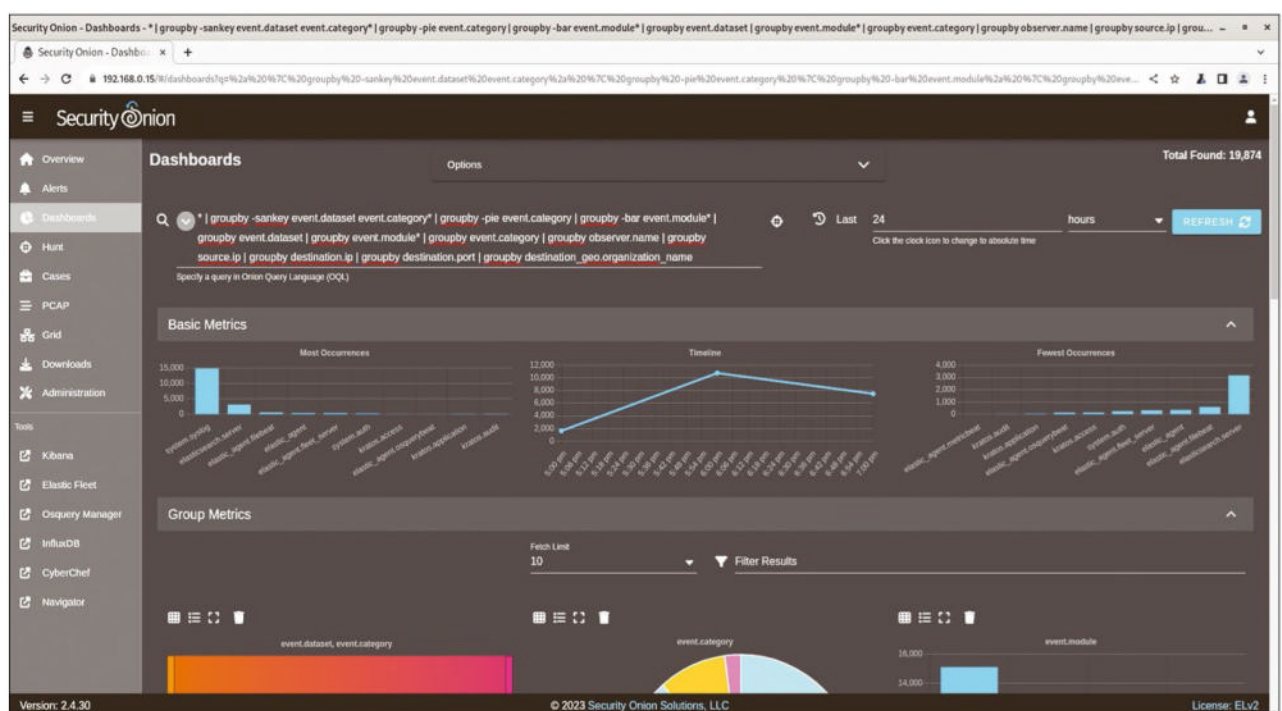


Figure 3: The dashboard contains graphical evaluations of your data.



## On the Desktop

The desktop version [16] of Security Onion (still labeled experimental) can be installed from the ISO image. It uses a heavily modified Gnome desktop on Oracle Linux 9.2. Besides the Chromium web browser, it only offers Wireshark and NetworkMiner as graphical tools for forensic work on the network.

You can set up the desktop version independently of the standard version. It only requires a minimum 50GB of free space on the local mass storage device. You can also install the desktop manually by typing

```
sudo so-desktop-install
```

at the prompt after completing the system configuration. The required packages are then preconfigured in your Security Onion installation. After restarting and authenticating, you are taken to the graphical desktop where you can access the Security Onion Console in a web browser. The two graphical tools already mentioned are also available.

## Conclusions

Security Onion is a powerful tool for data analysis and intrusion detection on the network. That said, Security Onion's complex structure means that the suite requires substantial hardware resources; deployment only makes sense in larger IT infrastructures. The developers therefore explicitly recommend purchasing new hardware for the security suite and offer their own appliances for customized application profiles in a web store. Getting started with the system is quite complicated; Security Onion is not something for hobby admins. Although the extensive and very detailed documentation flattens out the learning curve, it will still take you some time. The rapid release cycle of new versions is also a point of criticism. The project publishes more-or-less complete versions on its GitHub page virtually every week, typically prompting the need for hotfixes just a few days later. It would make more sense for the developers to test their software more thoroughly before releasing it and to avoid annoying users with images and scripts that do not work properly. ■

## Info

- [1] Security Onion: [\[https://securityonionsolutions.com/\]](https://securityonionsolutions.com/)
- [2] Product overview: [\[https://securityonionsolutions.com/software\]](https://securityonionsolutions.com/software)
- [3] Wazuh: [\[https://wazuh.com\]](https://wazuh.com)
- [4] OSSEC: [\[https://www.ossec.net\]](https://www.ossec.net)
- [5] Osquery: [\[https://www.osquery.io\]](https://www.osquery.io)
- [6] Beats: [\[https://www.elastic.co/beats\]](https://www.elastic.co/beats)
- [7] OpenCanary: [\[https://opencanary.readthedocs.io/en/latest/\]](https://opencanary.readthedocs.io/en/latest/)
- [8] Stenographer: [\[https://github.com/google/stenographer\]](https://github.com/google/stenographer)
- [9] Strelka: [\[https://target.github.io/strelka/#/\]](https://target.github.io/strelka/#/)
- [10] Zeek: [\[https://zeek.org\]](https://zeek.org)
- [11] Suricata: [\[https://suricata.io\]](https://suricata.io)
- [12] Kibana: [\[https://www.elastic.co/kibana\]](https://www.elastic.co/kibana)
- [13] Grafana: [\[https://grafana.com\]](https://grafana.com)
- [14] Download: [\[https://github.com/Security-Onion-Solutions/securityonion/blob/2.4/main/DOWNLOAD\\_AND\\_VERIFY\\_ISO.md\]](https://github.com/Security-Onion-Solutions/securityonion/blob/2.4/main/DOWNLOAD_AND_VERIFY_ISO.md)
- [15] Hardware requirements: [\[https://docs.securityonion.net/en/2.4/hardware.html\]](https://docs.securityonion.net/en/2.4/hardware.html)
- [16] Desktop variant: [\[https://docs.securityonion.net/en/2.4/desktop.html\]](https://docs.securityonion.net/en/2.4/desktop.html)

## Author

Erik Bärwaldt is a self-employed IT admin and technical author living in United Kingdom. He writes for several IT magazines.



## Azure Application Gateway load distribution tool

# Sharing the Load

In the Azure cloud, Microsoft offers the Azure Application Gateway managed service as a Layer 7 load balancer that needs virtually no internal resources to set up and operate. By Guido Söldner and Constantin Söldner

**Load balancers** are a central component for operating web applications – for both VM- and container-based applications. However, their legacy character poses a number of challenges for admins, particularly in terms of operation and the required expertise. If the application is to meet requirements in terms of high availability and scalability, for example, a number of prerequisites must be met. Depending on the criticality, several data centers are required, with management and monitoring tools to match.

A Layer 7 load balancer that works at the HTTP level is typically used for hosting web applications. However, this technology is also exposed to the challenges described above. Microsoft is looking to simplify this in Azure with the Azure Application Gateway (AAG). To avoid confusion, load balancers in Azure include:

- Azure Load Balancer, a Layer 4 service that works for TCP and UDP applications;
- Azure Front Door, a global Layer 7 load balancer particularly suitable for users from different geographical regions; and
- Azure Traffic Manager, a DNS-based service that directs users to a specific back end according to latency, location, and other criteria.

## Architecture

The Azure Application Gateway is Microsoft's standard for Layer 7 load balancing and supports the HTTP, HTTPS, and HTTP/2 protocols. However, the provider also advertises AAG as an application delivery controller that offers security functions (e.g., protection against distributed denial-of-service (DDoS) attacks).

To see how AAG works, refer to **Figure 1** for the individual components. The front-end IP is, as the name suggests, the IP address on which the application gateway listens to incoming requests. Depending on the application, a public and a private IP address or just a public IP address can be configured. However, only one private and one public IP address is available for each AAG.

A listener waits for incoming connections for a specific combination of port, protocol, hostname, and IP address. You can choose between a *Basic* and *Multi site* listener. Only one application per port is supported with the basic version, but the multisite listener lets you run several web applications with different hostnames on one port. If you want the listener to listen on HTTPS, an SSL certificate is required.

The rule determines the back-end pool to which requests are routed. The request-routing rule links a specific web application with a specific back end; AAG supports back ends in both the Azure Cloud and locally. You can specify Azure network interface controllers (NICs) and public and internal IP addresses as the back end, which can be physical and virtual machines, as well as containers. Azure Virtual Machine Scale Sets (VMSSs) and managed services such as Azure App Service are also supported. In terms of ports, all ports between 1 and 64,999 are theoretically available for the front end, with the exception of port 22 (SSH).

The HTTP setting is part of the request-routing rule and specifies the connection details of the back-end systems, including the port to be used, the protocol (HTTP or HTTPS), and the settings for connection draining and health checks. Connection draining allows a back-end instance to be removed as a load-balancing target, but in a way that lets it continue to support the existing connections. A custom probe is a user-defined

health check that the AAG runs against back-end instances. To classify an instance as functional, a health check sends regular requests to a user-specified URL and monitors whether or not the back-end instance sends the expected HTTP response (e.g., a status code of 200 or a specific character string in the response body).

## Connecting Applications

To host multiple applications, you do not need to provide multiple application gateways; rather, you can position multiple applications behind a single AAG, which is particularly relevant for microservice-based architectures that divide application components into different services. URL-based routing is often used, with the URL deciding which back end processes a request.

Besides URL-based routing, routing decisions can also be based on the hostname. To host several domains behind the same port number, you need to configure two multisite listeners. In this case, the two URLs resolve

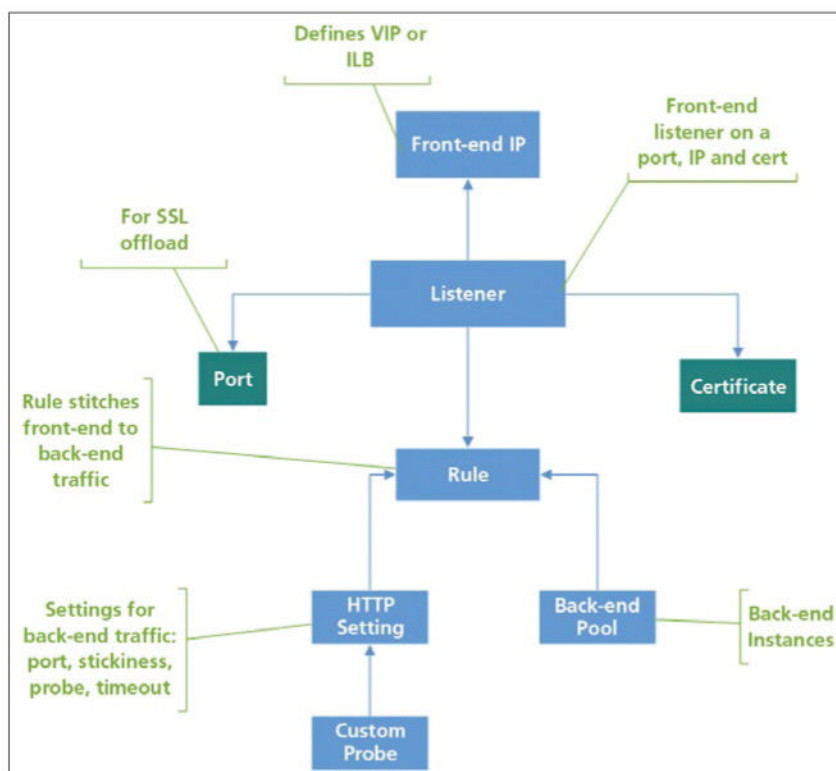
to the same IP address, but the application gateway can differentiate between them by the host header.

## High Availability and Scalability

The high availability and scalability of the load balancer are important aspects when you are hosting business-critical applications. Because the Azure Application Gateway is a managed service, you don't have to worry about having enough instances available. Azure makes life easy because you only need to specify the number of AAG instances and their availability zones (AZs). An AZ corresponds to an Azure data center location that is isolated from other AZs. The Azure Germany West Central (Frankfurt) region currently comprises three such zones, meaning that instances are available in up to three different locations for one AAG; however, for applications without special high-availability requirements, you could also operate the AAG in a single AZ only.

In total, an AAG allows up to 125 instances, which would probably be oversized for a single web application in the vast majority of cases. This high limit is also intended for lining up several applications behind a single AAG (multisite hosting). If one instance fails, Azure automatically provides a replacement. Azure also ensures that the individual instances are located in different fault and update domains.

Azure addresses the scaling of the application gateway. You can enable autoscaling wherever Azure provides additional AAG instances (or deletes them again) as a function of the workload. As with legacy autoscaling of virtual machines, you configure a minimum capacity and a maximum capacity (i.e., the minimum and maximum number of load balancer instances). Azure automatically scales up or down within these limits, which makes it possible to react to peak resource requirements while saving money whenever less capacity is required.



**Figure 1:** Azure Application Gateway architecture at a glance. ILB, internal load balancer; VIP, virtual IP; SSL, secure sockets layer.



A holistic scaling concept does not just include the load balancing layer, but also the back ends. Many back-end types come with scaling mechanisms by default, but you have to take care of this property yourself with others. The standard mechanism for scaling virtual machines in Azure is the VMSS, which scales the virtual machines up or down as a function of a selectable metric. In addition to metric-based autoscaling, scaling can also be based on a schedule or, in the case of predictive scaling, based on observed utilization patterns in the past.

If you use containers as back ends in an Azure Kubernetes cluster (AKS), you also have the option of scaling there. For containers that run on Kubernetes, the HorizontalPodAutoscaler or third-party tools, such as a Kubernetes-based event driven autoscaler (KEDA), can be used. You will also benefit from autoscaling if you use the Azure App Service to run web applications.

## Encryption of Application Traffic

Traffic encryption (SSL and TLS) is a standard requirement for modern applications. Because opening SSL connections entails some performance overhead, the question arises as to who should be responsible for SSL termination. Three approaches are possible:

- End-to-end encryption without termination on the load balancer: The connection between the client and the web application is encrypted end to end and is only terminated on the web server (pass-through). This variant is not supported by AAG but requires you to use regular Azure load balancers that operate in Layer 4. Another disadvantage of pass-through is that the load balancer cannot decrypt the headers sent with the HTTP request and therefore cannot make routing decisions from the headers. Therefore, the load balancer cannot distinguish between different services.

- Termination (SSL and TLS) on the load balancer without re-encryption: The connection is only encrypted between the client and the load balancer, whereas the connection between the load distributor and back-end instances remains unencrypted (SSL and TLS termination takes place on the load balancer). This variant is a good choice from a performance perspective because the back-end servers no longer have to worry about SSL and TLS termination and can focus on delivering the content, which requires installing an SSL certificate on the load distributor.

- Termination (SSL and TLS) on the load balancer with re-encryption: The connection between the client and the load balancer is encrypted, and a new TLS session is opened between the load balancer and the back-end instances. This approach offers greater security and is also a regulatory requirement in certain sectors. In this case, an SSL certificate is required both on the load balancer and on the back-end instances.

## DDoS Defense with Integrated WAF

DDoS attacks are an ever-increasing threat to web applications that are accessible on the Internet. The aim of such attacks is to bring a system to its knees because of a large volume of requests making the system inaccessible for legitimate tasks. DDoS attacks grow in number and size every year, prompting many companies to introduce appropriate defense mechanisms. One of the most important tools is web application firewalls (WAFs), which offer protection against attacks on Layer 7 (HTTP). A WAF typically fends off common threats such as SQL injection or cross-site scripting. Additionally, the AAG WAF can also use the IP reputation ruleset to block IP addresses that have already been the subject of frequent attacks. The Azure Application Gateway sits in front of the application and

processes incoming requests first, so it makes sense for it to act as a WAF at the same time. Azure offers a dedicated WAF that integrates with AAG. You need to select the WAF or WAF V2 tier when creating the application gateway. Note that an extra charge is levied for the additional WAF functionality: The costs are addressed in more detail in a moment. To protect applications, a WAF requires matching rulesets, which it uses to analyze incoming HTTP requests. Azure WAF is based on the OWASP Core Ruleset (CRS), but you can also create your own policies.

When configuring the WAF, you can choose between detection mode and prevention mode. In detection mode, the incoming requests and the potential decision about whether a request represents a threat are only logged, but no request is blocked. This mode is intended, say, to ensure that the WAF does not identify too many false positives (i.e., requests that are categorized as threats despite being legitimate). If prevention mode is active, the WAF sends a *403 The request is blocked* response if a request appears threatening.

## Load Balancing in Kubernetes

Azure Application Gateway can also handle load balancing for containers in Kubernetes (Kubernetes Pods). However, the required configuration does not take place directly on the gateway; rather, you create an Ingress object directly in the Kubernetes cluster. An Ingress is a Kubernetes API resource that you specify in the form of a YAML file.

You would typically use the `kubectl` command-line tool to create the Ingress object. An Ingress Controller is required for the load balancer to take over the routes specified in the ingress; this controller is a Kubernetes application that monitors the Kubernetes API for newly created ingresses. As soon as a new ingress is created, the Ingress Controller configures the load balancer with the information from it. You need

the Application Gateway Ingress Controller (AGIC) for AAG. If you use the Azure Kubernetes Service, the Ingress Controller can be installed with an add-on; alternatively, it can also be installed by a Helm chart.

## Costs of the Application Gateway

The price for AAG depends on whether you use v1 or v2. However, v1 is already considered legacy, so you are only looking at the costs for version 2 in this case. Additionally, as already mentioned, you pay more if you commission the WAF functionality. The price comprises two components: a fixed price per hour and a variable component (capacity units per hour). In the case of the App Gateway without WAF, the fixed share is \$0.20/gateway hr;

with WAF, the cost is \$0.36/gateway hr. The hourly charge per capacity unit is \$0.008/hr; with WAF it is \$0.0144/hr.

The capacity units are somewhat more complex to calculate because they comprise three parameters: 2,500 persistent connections, 2.22Mbps throughput, and a compute unit, which in turn depends on other calculation operations. If one of these three parameters is exceeded, an additional capacity unit is billed. The price model shows that it takes some experience to be able to estimate the costs of the App Gateway accurately.

## Conclusions

Compared with other load balancers, the Azure Application Gateway does not offer the same number of features and functionality. However, if you do

not have any specific requirements, AAG is a solid load distribution tool and is well integrated into the Azure world – especially when it comes to integration with Azure Monitor. At the end of the day, Azure App Gateway's big advantage is that it is a managed service, which means to set up and operate it, you need virtually no internal resources. ■

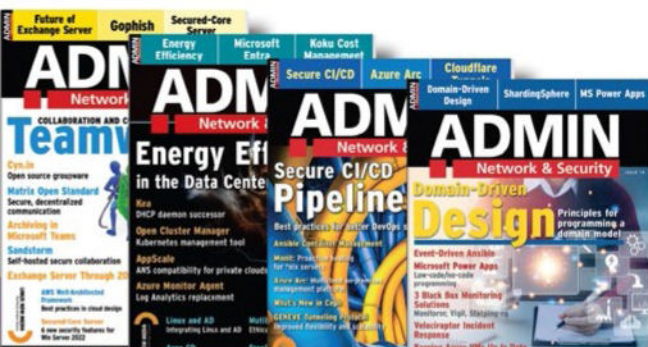
### Authors

**Guido Söldner** is a Principal Consultant at Söldner Consult. His subject areas include cloud infrastructure, automation, DevOps, Kubernetes, machine learning, and enterprise programming with Spring. He has more than 10 years experience with cloud computing.

**Constantin Söldner** is a Principal Consultant at Söldner Consult. He specializes in Cloud Computing and has a strong background in Kubernetes, DevOps, and automation. ■

# What?!

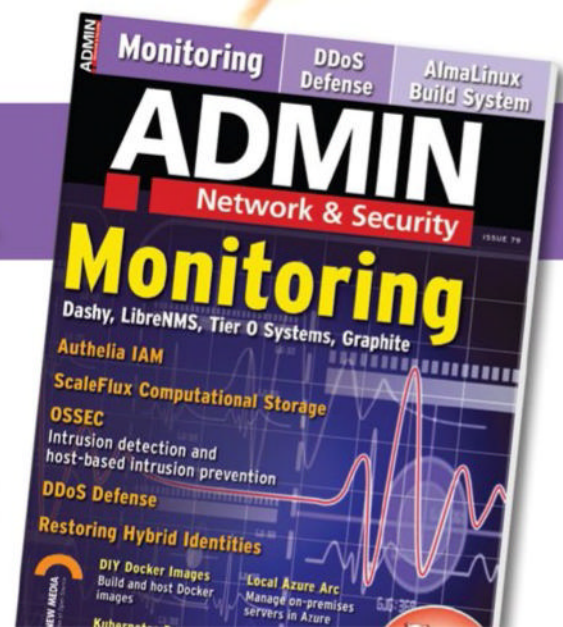
## I can get my issues SOONER?



Available anywhere, anytime!

Sign up for a digital subscription to improve our admin skills with practical articles on network security, cloud computing, DevOps, HPC, storage and more!

Subscribe to the PDF edition:  
shop.linuxnewmedia.com





## Chatbots put to the scripting test

# Beginner's Mistake

The AI skillset is currently limited, so you don't yet have to worry about AI replacing programmers. We look at the capabilities of AI scripting with free large language models and where it works best. By Andreas Stolzenberger

**A large language model** (LLM) uses what is known as a transformer architecture, hence the name “generative pretrained transformer” (GPT). Put simply, an LLM represents each word used in a sentence as a series of mathematical vectors. The trained model comprises several layers of transformer encoders and decoders that put the vector groups of words from a sentence or word block in a suitable context.

As the model learns, it creates encoders that suggest a connection between the vectors of terms (e.g., “ship” and “water” or “helicopter” and “flying”). During the learning phase, the LLM tries to build correct sentences. It compares these with the correct answers given by the trainer. The difference between the LLM's response and the correct response is transferred to the transformer layers and optimizes the LLM's function. A generic LLM therefore becomes proficient in one

or more languages with which it has been trained and has knowledge of the data used in the training.

For example, a model such as ChatGPT 3.5 was trained with 175 billion parameters – its state of knowledge dates back to September 2021. GPT-4 already has more than a trillion parameters. After completing the basic training, an LLM can be improved with further levels of knowledge. These difference models are known as a low-rank adaptation of large language models (LoRAs). For example, a LoRA can help a generic language model learn how Python programming works without having to retrain the entire LLM.

This mode of operation also shows the weaknesses of LLMs: They are not creative and do not generate any new information. They only use existing knowledge and reformulate it to match the question. Of course, they have a massive amount of information that

no single person has at their disposal, but the LLM is again limited to the information with which it was trained.

## Model Weaknesses

The limitations of the training data are the first concrete weakness of an LLM. ChatGPT 3.5, for example, was trained with a database from 2021 and cannot answer questions about later events. If you use ChatGPT 3.5 to create Python code, for example, you will be given snippets that are compatible with Python 3.8. The LLM does not take into account the changes in Python 3.11 from October 2022. Although this might not be a major issue with Python, it has a far more serious effect on languages such as Ansible, which has undergone many changes between versions 2.9 and 2.15.

Another problem regarding the future is that, at the moment, vast quantities of new text are being created by

Lead Image © lexandersikovi, 123RF.com



LLMs with their outdated knowledge. However, the text is not flagged as being LLM-based and derivative of others' work but is being sold as original content. Some of this content contains outright lies, because it is derived from hoaxes, such as the fake *World of Warcraft* feature perpetuated by a few fans [1]. As amusing as this story is, it also means that this false information will end up in the training data of future LLMs. After all, no one can manually sift through and qualify a billion parameters before they approve them for use in training. Future generic LLMs will therefore be worse rather than better. Although the quantity of training data is growing rapidly, its quality is continuing to decline, meaning that the quality of future LLMs will likely to be worse. The real future of LLMs instead lies with small models that are trained with a well-qualified and fairly private database.

An often overlooked but serious weak point relating to the tools used by LLMs is that vector mathematics should always provide the exact same answer to a question – namely, the one with the highest probability of coming back from the transformers. Imagine if image-generating models such as DALL-E, Midjourney, or Stable Diffusion always provided the user exactly the same image when requesting an astronaut riding a donkey. The technology would be too precise and therefore boring. Therefore, all image and text models add a seed to the user questions. A seed is nothing more than a large random number, which means ChatGPT processes the request supplemented by one roll of the dice in the transformers to produce a response. However, in the case of generating program code, you would want the mathematically most probable result, not one that is co-determined by a random generator.

## Generating Code with LLMs

A number of LLMs are available to generate code free of charge. I start with ChatGPT [2], followed by IBM watsonx Code Assistant for Red Hat

Ansible [3] and the self-hosted Oooba-Booga [4]. Watson Code Assistant is currently only available free of charge for Ansible (Ansible Lightspeed), but the other two options deliver program code in PowerShell, Python, Bash, or Perl.

The free version of ChatGPT used for this article means I was restricted to version 3.5. Anyone working with ChatGPT must always bear in mind that OpenAI stores all user input and uses it to train future LLM generations, so make sure you don't feed the chatbot confidential information.

In the first test, ChatGPT created a Bash script that listed all RPM packages installed on a Linux distribution for which updates were available.

This task was quite simple, but I wanted output that listed the package name, the currently installed version, and the available version. A human-programmed script for this task might look like Listing 1.

In the script, `dnf list updates` returns the available updates and shows the name of the package (pn), the available package version (pv), and the repositories (pc, which is not needed). Because `dnf` does not reveal the currently installed package version, the script uses `rpm -q`, followed by a cosmetic clean-up of the available version (pv3), which removes the name of the distribution from the update version number. The first two output lines without package names from the `dnf list updates` line are filtered out by the `if` statement.

Because hardly any administrator needs the detailed version number function, no demo scripts exist for this special case on the Internet and therefore no ready-made training data for a model. When asked to *Write a bash script that lists RPM packets that can be updated; the list must contain the name of the packet, the currently installed version number, and the version number of the available update,*

Listing 1: List RPM Packages

```
#!/bin/bash
printf "%-40s %-20s %-20s\n" "Package Name" "Update" "Installed"
dnf list updates | while read pn pv pc;
do
    if [ "$pn" = "Last" ] || [ "$pn" = "Available" ]; then continue
    fi
    pv2=$(rpm -q --qf '%{VERSION}\n' $pn )
    IFS='- ' read -ra pv3 < "$pv"
    printf "%-40s %-20s %-20s\n" "$pn" "$pv3" "$pv2"
done
```

ChatGPT did not provide a working script: The bot suggested the use of `updates=$(dnf check-update)` to determine the packages that could be updated. So far that suggestion is correct, but ChatGPT claimed that the command output would display the package name and the installed and available version in three columns. This statement is not true: The second column contained the available version and the third the name of the repository. In response to the complaint (*This does not work. \$(echo "\$line" | awk '{print \$3}')* returns the name of the packet repository, but not the available version), ChatGPT responded (*I apologize for the oversight. You're correct; the 'dnf check-update' output includes repository information. Here's an updated script that correctly extracts the available version*) and then returned another script that still didn't work, because it still split up the output line in the wrong way (Figure 1).

In the next step, ChatGPT received the working, human-generated script for analysis and provided a very precise explanation, so I asked ChatGPT if it could improve this script, to which the bot responded: *The script you provided is functional, but it can be simplified and improved in terms of readability and efficiency. Here's a revised version of the script.* The script dispensed with the `while` loop and instead suggested a somewhat more complex `awk` call. In principle, this solution is a good idea, but the script did not work because it did not implement the `Awk` syntax correctly and proposed some fairly dubious variable filters.

In the next attempt, I wanted the chatbot to create the desired functionality in Ansible instead of Bash. This attempt also failed – quite badly. The very first task of the playbook it offered was fatally flawed:

tasks:

```
- name: Update package cache
  package:
    name: "*"
    state: latest
    become: yes
```

If the `ansible.builtin.package` module starts with `name: "*" and state: latest, it does not update the package cache as claimed, but updates all updateable packages directly. The rest of the playbook is then useless. If Ansible executes command: dnf list updates after this unintentional full update, there is no response because all of the installed packages are already up to date.`

The other ChatGPT suggestions were not helpful either. Other tasks in the playbook used modules such as `command: or set_fact:, which experienced`

Ansible programmers tend to avoid wherever possible. Unfortunately, the Internet is full of bad examples of Ansible programming, which means the suggestions from the Internet-trained LLM are unlikely to provide particularly good code.

As a last attempt, I asked ChatGPT to generate some PowerShell code: *Create a PowerShell script for Windows that queries a username and password in a graphical dialog using forms. With that information, the script will create a Windows user.* You can find a whole series of examples of this query online. Logically, ChatGPT came up with a working script for this task that looked more or less like the demo scripts from various websites. However, any administrator could have found this code with the use of a search engine, without resorting to an AI bot.

The bottom line on ChatGPT is that if you want to program something that other users have created and published before September 2021, feel free to ask ChatGPT. However, you could also use a search engine to

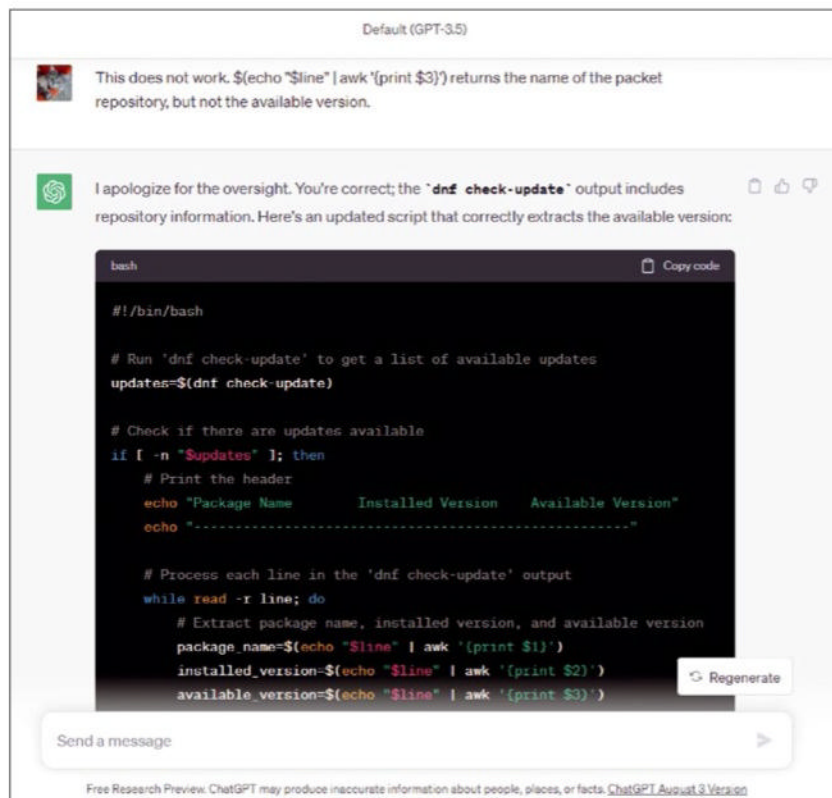
find suitable sources, and you would better be able to judge the trustworthiness of these sources. Be more cautious with special requests and, above all, check the code suggested by the chatbot in detail.

## Code with Ansible Lightspeed

The aptly named IBM Code Assistant, a suite of generative AI-assisted products, provides several purpose-built models that, unlike a generic LLM, have been trained for specific tasks. Thankfully, IBM indicates with the Code Assistant name that it supports programmers and does not generate code with the aim of replacing people. A paid version is intended to help programmers rewrite the outdated Cobol programs on mainframes in Java, among other things.

The watsonx Code Assistant for Ansible is better known as Ansible Lightspeed and is currently available to all users free of charge. However, the manufacturer has not yet decided whether and for how long this picture will remain the same. The tool is part of the Ansible plugin for the Visual Studio Code editor. You only have to link a GitHub account with the Ansible plugin to use it. As with ChatGPT, the user of the free version has to agree that their code can be used by the manufacturer to improve the model. The same applies to Lightspeed. Only use it for code that does not contain personal or confidential information.

Lightspeed only supports Ansible as a language, but unlike ChatGPT 3.5, it has up-to-date knowledge of Ansible 2.14. Users do not integrate Lightspeed by sending a chat request. Instead, as an Ansible developer, you write your code and Lightspeed suggests code blocks on the basis of code already in use and the name of the tasks. The results are fairly mediocre, especially at the beginning of an Ansible playbook, because Lightspeed has to guess what the user wants to do from the name of the first task. The more code you write, the better the Code Assistant suggestions become; it then also



**Figure 1:** After criticism, ChatGPT apologized for the faulty code and provided a revised script, which still contained the criticized errors.

inserts previously declared variables in the correct places.

Lightspeed did not provide any working code for my example off the cuff, but if you then continue to write a significant part of the automation yourself in your playbook, you will start to receive correct suggestions for further tasks. These suggestions then match your existing code and follow the syntax of the current Ansible version.

As an assistant, Lightspeed supports users creating longer playbooks or roles. Tasks are suggested that match the existing playbook programming and the variables used there. However, do not expect watsonx to conjure up complete playbooks out of a hat as an LLM in the style of ChatGPT tries to do.

## Local AI with OobaBooga

OobaBooga is an open source project that generates a simple web interface for LLMs. It runs on all common operating systems with a suitable Python interpreter. You can ask it anything because no data is transmitted to a provider on the Internet. If Miniconda, the leaner version of the popular Anaconda Python distribution, is not already available globally on your system, the simple one-click installer sets it up.

For the web user interface (UI) to answer questions in the style of ChatGPT, it also needs one or more LLMs, which must be quantized to work on regular PCs. Put simply, quantization compresses the model by reducing the accuracy of the vectors. In principle, two quantizations are suitable for use on the PC: GGML for models that work on the CPU, and GPTQ for GPUs (NVIDIA).

The most popular website for open source models is Hugging Face [5], which now has hundreds of freely usable LLMs of different quality for different areas of application, many of which are merges (i.e., models that combine several existing LLMs with different weightings to create a new model). In addition to LLMs with optimization for chat or text creation

(Storytellers), “coder” models generate program code.

Hugging Face user “TheBloke” offers a large number of ready-made quantized models in his repository that work on standard PCs. However, their performance leaves much to be desired. To come to grips with the technology, you need a computer with a powerful NVIDIA GPU. The VRAM capacity is particularly important because a GPTQ model must fit completely into the GPU’s memory. To be able to load a large LLM, your GPU must have 24GB of VRAM. That said, specialist coding models will get by with less because they can do without the general knowledge of the chat or storyteller LLMs.

For this article, I used the CodeUp-Llama-2-13B code generation model [6], which makes do with the 12GB of VRAM provided by the RTX 4090 in the lab computer. Again, I asked the coding model to generate the Bash script in Listing 1. Unlike ChatGPT, the self-hosted LLM initially provided a working approach by suggesting `rpm -qa` and `rpm --queryformat` as tools to determine the versions; however, it then messed up the Bash syntax and, later in the code, started using variables that it did not declare previously. Code generation for Ansible suffered a similar fate. Again, the first lines of the proposed playbook made perfect sense, but as the output progressed, the LLM produced confused output and mixed Ansible syntax with Bash syntax.

The reasons for degenerating code quality are probably the high degree of quantization and the limited token buffer, which is the LLM’s memory. Put simply, if the buffer overflows during the LLM’s response, the model forgets what the user asked in the first place.

I looked at other open models, including models that have not been optimized for code generation. All models started off on the right path, but then all failed in terms of syntax and output values. Most of the proposed `| grep` and `| awk` filters do not work. All of the Ansible code generated by generic models was useless.

Self-hosted open source LLMs on your own graphics card provide some surprisingly good answers, at least as long as you chat with them casually and do not ask for working program code. The quality of the quantized models and the limited token buffers simply are not up to the task right now.

## Conclusions

As always, you’ll find far more hype than reality when it comes to AI-generated scripts. The current tools cannot reduce your scripting workload. At best, they can provide some assistance. In the future, the models will certainly grow and become more powerful, but it is doubtful whether they will also become better because of the degenerating training data. What will therefore be of particular interest to application developers is helper LLMs with limited, but specialized, databases. ■

### Info

- [1] Gamers fool AI news sites: <https://www.polygon.com/23803152/world-of-warcraft-glorbo-ai-news-site-reddit>
- [2] ChatGPT: <https://chat.openai.com/auth/login>
- [3] IBM watsonx Code Assistant for Ansible: <https://www.ibm.com/products/watsonx-code-assistant>
- [4] OobaBooga: <https://github.com/oobabooga/text-generation-webui>
- [5] Hugging Face: <https://huggingface.co>
- [6] CodeUp-Llama-2-13B: <https://huggingface.co/TheBloke/CodeUp-Llama-2-13B-Chat-HF-GGML>

### The Author

**Andreas Stolzenberger** worked as an IT magazine editor for 17 years. He was the deputy editor in chief of the German *Network*

*Computing* magazine from 2000 to 2010. After that, he worked as a solution engineer at Dell and VMware. In 2012 Andreas moved to Red Hat. There, he currently works as principal solution architect in the Technical Partner Development department.







## MySQL upgrade obstacles

# Stumbling Blocks

A number of breaking changes have been introduced between MySQL 5.7 and 8.0. We show you how to navigate this mandatory upgrade. By David Berube

**Benjamin Franklin famously said,** "... In this world nothing can be said to be certain, except death and taxes." From this quotation, we can determine that he was not, in fact, a system administrator; if he were, he'd have added "software upgrades" to that list.

MySQL is no exception to Benjamin Franklin's famous quote. After all, on October 21, 2023, MySQL 5.7 entered end-of-life (EOL) status. Consequently, no patches or updates will be available from official sources, and although MySQL 8.0 has been in General Availability (GA) status since 2018, many users have not yet upgraded. Since you're reading this article, you may very well be among them.

Fortunately, with some planning, you can take the sting out of this particular mandatory upgrade. I won't be able to discuss every breaking change or possible issue you might encounter, so be sure to follow the official MySQL instructions.

To begin, I'll discuss which systems can upgrade to MySQL 8.0 and what you can do if you aren't eligible.

## Who Can Upgrade?

The upgrade to MySQL 8.0 or later is only possible from the 5.7 General Availability (GA) releases, not release candidates or other development releases. The earliest GA release is 5.7.9; versions of 5.7 earlier than that will have a suffix (e.g., "5.7.8 rc" for the release candidate or "5.7.1 m11" for milestone 11).

Likewise, upgrades from versions before the 5.7 major release are not supported. So, for example, if you have a MySQL 5.6 installation still running, you will first need to upgrade to MySQL 5.7. After that, you can upgrade an installation to 8.0. Generally speaking, MySQL only officially supports upgrades across one major release, so if you want to upgrade from, say, 5.7 to 8.2, you'll first have to upgrade your MySQL 5.7 installation to MySQL 8.0, then 8.1,

and finally 8.2. It's worth noting that MySQL 8.0 goes EOL in April 2026, which is not terribly far in the future. MySQL 8.0 is a long-term support (LTS) release, however, which means you won't get much additional time by upgrading past that – at least until 8.4 is released, which is scheduled to be the next LTS release.

In any event, minor upgrades such as from 8.2.0 to 8.2.1, can be done several at a time. If, for example, you're on an older version of MySQL 5.7, there's no need to upgrade to each minor version in between individually; you can skip minor versions as needed. Officially, you can upgrade from any GA release of MySQL 5.7 to 8.0, although some have preferred to upgrade to the latest MySQL 5.7 release before attempting an upgrade to 8.0.

It's also important to note that a major breaking upgrade, such as that from MySQL 5.7 to 8.0, is a good time to evaluate your choices in technology and vendors carefully. For example, if you're considering changing operating

Lead image © cycloneproject, i23RF.com

systems, architectural changes such as moving from on-premises to cloud or vice versa, and so on, now might be a good time to schedule that move. You're likely going to have to schedule significant testing and downtime, so there may be an efficiency benefit to doing both at once.

Likewise, if you're considering changing database technologies, either to an entirely different family such as PostgreSQL or to something closely related like MariaDB, the upgrade from 5.7 is a good time to do that.

## Potential Upgrade Issues

One significant change involves correlations and encoding. The default collation and encoding for MySQL 5.7 was `latin1`, meaning that when a `CREATE TABLE` statement is run, it creates the new table with the `latin1` character set and the `latin1_swedish_ci` collation. Many installations, however, are on UTF8 encoding, which is the new default on MySQL 8.0.

The default meaning of UTF8 is different under MySQL 5.7 and MySQL 8.0. On MySQL 5.7, `utf8` is interpreted as `utf8mb3`, whereas under MySQL 8.0 it is `utf8mb4`. Although `utf8` and `utf8mb3` are largely compatible – with `utf8mb4` supporting more characters – it is worth checking to see whether your applications use `utf8mb3`, `utf8mb4`, or `latin1`, and it might be a good time to standardize all of your collation and encoding settings. The `utf8mb4` choice would be excellent for most situations. An unexpected shift from `latin1` to `utf8mb4` might cause data imports or other processes to fail, so setting the encoding specifically to `utf8mb4` wherever possible may be a good move. Both MySQL 5.7 and 8.0 support all three encodings, so you can change your databases and tables to `utf8mb4` ahead of time to avoid a surprise caused by a change in defaults. If you want to keep the old defaults when you upgrade to MySQL 8.0, you can add the lines

```
[mysqld]
character_set_server=latin1
collation_server=latin1_swedish_ci
```

to the `my.cnf` file before you upgrade, because those settings are supported by MySQL 5.7, as well. Conversely, you can add the lines

```
[mysqld]
character_set_server=utf8
collation_server=utf8mb4_0900_ai_ci
```

to your MySQL 5.7 configuration file to make its behavior match that of 8.0.

## Removed Temporal Data Types

Some data types, or at least some variants of data types, have been removed from the new version of MySQL. So-called old temporal data types are no longer supported. These data types only have second precision. Now, newly created tables (i.e., any tables created after MySQL 5.5) will automatically have fractional second precision; under normal circumstances, the 5.6 upgrade will automatically upgrade some types, but under some circumstances they can still exist.

The MySQL documentation helpfully provides these two queries to detect temporal columns of the old type:

```
SET show_old_temporals = ON;
SELECT table_schema, table_name,
       column_name, column_type
FROM information_schema.columns
WHERE column_type LIKE 'timestamp
/* 5.5 binary format */'G
```

If you find any, you can fix it with the command

```
ALTER TABLE <some_table> FORCE;
```

which rebuilds the table, possibly taking a good deal of time. You can also dump the table as an SQL file, recreate it, and then reload from a backup, which should recreate the columns as new, more precise columns, as well.

## Naming Issues

Some foreign key constraints from prior versions of MySQL can be incompatible because of excessive

length. Specifically, a partially new constraint is that foreign key names cannot exceed 64 characters; in general, the limit was already 64 characters; however, in some cases, InnoDB would generate longer foreign key constraint names, typically as a result of long table names in non-English languages with multibyte characters. In a similar vein, before MySQL 8.0, views could have column names up to 255 characters; however, to unify column name restrictions, explicit column names should only be 64 characters long. It's unlikely most installations will experience this problem, but if you do, the automated MySQL upgrade check scripts I discuss later should catch the issue.

As has been the case with prior major version upgrades, the number of reserved words has increased. Note that simply because a term is a reserved word does not mean you cannot have a column or table with that name. It simply means that to use that word you have to enclose it in backticks. Query generation tools such as ActiveRecord or SQLAlchemy automatically use backticks when appropriate; however, it may be wisest simply to avoid the use of such reserved words to eliminate the possibilities. Several new reserved words do have names that might be plausibly found as a column or table name (e.g., `active` and `admin`); the list of reserved words, which you can find in the MySQL documentation, is worth reviewing.

MySQL used to support ordering in the `GROUP BY` clause; MySQL 8.0 drops that support, so queries like

```
SELECT <...> FROM <table>
GROUP BY <something> ASC;
```

will need to be rewritten as:

```
SELECT <...> FROM <table>
GROUP BY <something>
ORDER BY <something> ASC;
```

## GRANT Statement Changes

In MySQL 8.0 the `GRANT` statement has much less functionality than it

did before. Previously, it could create users if they didn't already exist, and it could alter user metadata. Now, GRANT statements can only be used as the name implies: to grant privileges to already created users. If you have administration scripts that create users with a GRANT statement, you should rewrite these to explicitly use CREATE USER statements to avoid issues. Likewise, other changes to a user's metadata can be made with the ALTER USER statement. MySQL 5.7 has both of these statements already, so these changes can be made before the MySQL 8.0 upgrade process.

## Authentication Methods Break Older Clients

A very significant change was made to the default authentication method in MySQL 8.0. MySQL has pluggable authentication methods, so you can use different methods for different installations. The default method for MySQL 5.7 is called *mysql\_native\_password*. The default in MySQL 8.0, however, is *caching\_sha2\_password*. In some cases, this transition will be seamless. Pre-existing user accounts will not be automatically changed but will be updated to the new default when their passwords are changed. Newly created accounts will use the *caching\_sha2\_password* plugin. Some older applications might not understand how to interact with the *caching\_sha2\_plugin*. Unfortunately, such applications might fail when connecting to a server with a default authentication method of *caching\_sha2\_plugin*, even when connecting to a not-yet-updated user. Ideally, you would update such old applications. If this isn't an option, you can set the following option in *my.cnf* to re-enable the old plugin:

```
default-authentication-plugin=mysql_native_password
```

Note that this option still might not fix authentication issues if your users had been created during the period of time when the *caching\_sha2\_plugin* was

the default; you can manually adjust such users with statements such as

```
ALTER USER 'username'@'somehost' IDENTIFIED WITH mysql_native_password BY 'a_secure_password';
```

On a related note, the PASSWORD() function is no longer available in MySQL 8.0. If you have scripts that create users or change user passwords, they will likely need to be rewritten, as is also the case with the GRANT changes. For example, the code

```
SET PASSWORD FOR 'tom'@'bob' = PASSWORD('test');
```

can be rewritten as

```
ALTER USER 'jeffrey'@'localhost' IDENTIFIED BY 'new_password';
```

Likewise, if you're using the PASSWORD function for other purposes, you'll need to rewrite the query. Note that MySQL 8.0 does include cryptographic functionality, such as the SHA2 function, which returns a SHA2 hash of its input. You can likely replace non-MySQL authentication-related uses of the PASSWORD function with that.

## Upgrading Applications

A number of steps should be taken before upgrading MySQL from 5.7 to 8.0. You'll need to find out if your applications support 8.0. Some third-party applications might require upgrades. Fortunately, because MySQL 8.0 was released for general availability in 2018, most vendors should have long since updated their products – assuming, of course, that they haven't gone defunct. If you're regularly upgrading your third-party applications, you've likely already moved to a MySQL 8.0-compatible version. If you have any applications or scripts you've developed in-house, they will have to be checked, as well. If you're using a continuous integration (CI) tool, such as Jenkins or CircleCI, it might be wise to run your automated

tests twice – once with your legacy version of MySQL and once with 8.0 – so you will be confident that your code works with MySQL and stays working with MySQL until you perform the upgrade.

Before beginning the backup process, it is likely wise to take both a logical or physical backup, or both, of your MySQL database and attempt upgrading in a non-production environment. If it's not practical to test the entire database in this way, then a subset thereof can be used. Although performance in a test or development environment won't be identical to production, you might be able to detect errors attributable to differences in the environments.

Related to that concept, a very helpful tool called *pt-upgrade*, a part of the Percona Toolkit, is freely available and designed to compare two different database servers. For example, say you have a backup of your production database loaded onto two test machines, *test57* and *test80*. Furthermore, say you've downloaded some sample queries from your production machine's slow log into a file called *prod-mysql-slow.log*. You can then run *pt-upgrade*:

```
pt-upgrade h=test57 h=test80 prod-mysql-slow.log
```

This entry will run the commands in *prod-mysql-slow.log* on both servers, check for errors, check that both servers returned the same data, compare performance results, and more. If you get errors to queries on MySQL 8.0 but not 5.7, then you've likely been affected by a breaking change. The *pt-upgrade* tool can also run arbitrary queries from text files, from packet capture, and a lot more, and you can check the Percona Toolkit documentation for more details.

Note that *pt-upgrade* is designed to be run in test, not production, environments; it can't produce valid timing data if one or both servers are loaded, and if the data is changing while the process is running, the consistency checks will likely produce false positives.



## Upgrading Official Tools

Once you've established that the applications you run are compatible with MySQL 8.0, it's time to inspect the database itself thoroughly with one or both of two official tools. The older tool is `mysqlcheck`:

```
mysqlcheck -u root -pmy-secret-pw 2
--check-upgrade --all-databases
mysql.columns_priv      OK
mysql.db                OK
mysql.engine_cost       OK
mysql.event             OK
...
```

The MySQL Shell `checkForServerUpgrade` command is the newer tool. The official MySQL documentation mostly references the latter tool, but many recommend running both for the sake of completeness. The `checkForServerUpgrade` command is distinct from the much older `mysql` command-line tool and has to be installed separately. It does have a very similar purpose, but has considerably more features – notably including built-in JSON output for queries. You can install the MySQL Shell with either the `mysql-shell` Yum or Apt packages, which require you to have the official MySQL repositories enabled, or by downloading from `mysql.com`. Unlike the traditional `mysql` command, it's not included by default with MySQL server (Figure 1). Once installed, you can use the server upgrade check:

```
mysqlsh -e 'util.checkForServerUpgrade()'
```

You can also simply invoke the tool with the `docker` command:

```
docker run mysql:8 mysqlsh 2
-e 'util.checkForServerUpgrade()'
```

In both cases, you can add authentication options as needed through options (e.g., `-u` and `-p`) similar to the old `mysql` client; you can also use the new `uri` parameter (Figure 2):

```
mysqlsh 2
--uri=root:my-secret-pw@mysql57:3306 2
-e 'util.checkForServerUpgrade()'
```

```
mysql.proc              OK
mysql.procs_priv        OK
mysql.proxies_priv      OK
mysql.server_cost       OK
mysql.servers           OK
mysql.slave_master_info  OK
mysql.slave_relay_log_info OK
mysql.slave_worker_info  OK
mysql.slow_log          OK
mysql.tables_priv       OK
mysql.time_zone         OK
mysql.time_zone_leap_second OK
mysql.time_zone_name    OK
mysql.time_zone_transition OK
mysql.time_zone_transition_type OK
mysql.user              OK
sys.sys_config          OK
(base) → very important server
[0] 0:zsh* "Forgeofdata" 11:41 13-Mar-24
```

Figure 1: Example `mysqlcheck` output during a pre-upgrade check.

```
The following variables have problems with their set authentication method:

Warning: default_authentication_plugin - mysql_native_password authentication
method is deprecated and it should be considered to correct this before
upgrading to 8.4.0 release.

32) Check for deprecated or invalid authentication methods in use by MySQL
Router internal accounts.
No issues found

Errors: 0
Warnings: 6
Notices: 13

NOTE: No fatal errors were found that would prevent an upgrade, but some potential
issues were detected. Please ensure that the reported issues are not significant
before upgrading.
(base) → mysql-57-to-mysql80
[0] 0:zsh* "Forgeofdata" 11:33 13-Mar-24
```

Figure 2: Example `mysqlsh` output.

### Listing 1: `mysqlsh` Output

```
The MySQL server at mysql57:3306, version 5.7.44 - MySQL Community Server
(GPL), will now be checked for compatibility issues for upgrade to MySQL 8.3.0.
To check for a different target server version, use the targetVersion option...

1) Usage of old temporal type
No issues found

2) MySQL syntax check for routine-like objects
No issues found

...

9) Usage of obsolete sql_mode flags
Notice: The following DB objects have obsolete options persisted for
sql_mode, which will be cleared during the upgrade.
More information:
https://dev.mysql.com/doc/refman/8.0/en/mysql-nutshell.html#mysql-nutshell-removals

sakila.film_in_stock - PROCEDURE uses obsolete NO_AUTO_CREATE_USER sql_mode
sakila.film_not_in_stock - PROCEDURE uses obsolete NO_AUTO_CREATE_USER
sql_mode
sakila.get_customer_balance - FUNCTION uses obsolete NO_AUTO_CREATE_USER
sql_mode
sakila.inventory_held_by_customer - FUNCTION uses obsolete
NO_AUTO_CREATE_USER sql_mode
...
```

In either case, you should see output like that shown in [Listing 1](#). As you can see in this case, the old temporal date types discussed earlier are not present. The `NO_AUTO_CREATE_USER sql_mode` warning relates to the `GRANT USER` changes mentioned earlier in the article. Because `GRANT USER` is no longer allowed to create users, the `NO_AUTO_CREATE_USER sql_mode` relating to when `GRANT USERS` is and is not allowed to create users has been removed. Therefore, the above-mentioned objects will need to be rewritten.

Once the output of both tools is scrutinized and issues found are corrected, you can proceed to the upgrade. Immediately before upgrading, it is wise to take both a logical and physical backup, with the use of tools such as `mysqldumper` or `mysqldump` for the logical backup and a tool such as `xtrabackup` or another copying tool for the physical backup ([Figure 3](#)). Typically, as with all upgrades, backups will be scheduled for a period of low traffic, and, depending on your situation, a maintenance window is announced. Although in theory the process can be, as in the words of Oracle's website, "seamless," it is instead wiser to plan for a somewhat "seamful" experience instead.

## In-Place and Upgrade Options

The two main upgrade paths are in-place and with the upgrade command.

The in-place upgrade is likely the first method most admins consider. This path involves stopping the old server process, replacing it with a new server binary, and then starting the new binary. If you've installed MySQL manually, you can keep both binaries, but realistically, most systems have MySQL installed through a package manager, so I'll discuss that in a bit more detail.

For example, on a Debian-based system with the official MySQL repositories, the routine would be

```
sudo apt-get update
sudo dpkg-reconfigure mysql apt-config
sudo apt-get update
```

The second command prompts you to select a MySQL major version; after selecting 8.0 you then need to run the update command again. To stop the MySQL process, enter

```
sudo systemctl stop mysql
```

If you haven't already, now is a reasonable time to make the physical backup, because you can do so without the server running by simply copying the MySQL data directory. After that, you can now commence the upgrade process:

```
sudo apt-get install mysql-server
```

If you've decided to make any changes to configuration files (e.g., changing the default auth plugin, as mentioned earlier, or changing the

default character set and collation) now is a good time to do so. Now you can start the server:

```
sudo systemctl start mysql
```

At this point, the MySQL server should automatically upgrade your tables. Unlike earlier versions of MySQL, you do not have to run the separate `mysql_upgrade` tool. Taking an additional set of backups at this point is good practice. In the case of some unusual behavior, it will allow you to compare pre-upgrade and post-upgrade backups. Although you likely won't need to do this, if such information turns out to be valuable, you won't be able to get it any other way. If you're running a Debian-based system, but not from the official repository, the official recommendation is to change to using the repository before the upgrade. Documentation on how to do that is available on the MySQL website.

On `dnf`-based systems, instead of `apt-get`, you can use

```
sudo systemctl stop mysql-server
sudo dnf config-manager --disable mysql57-community
sudo dnf config-manager --enable mysql80-community
sudo dnf upgrade mysql-server
sudo systemctl start mysql-server
```

As before, I recommend manually stopping the process with `systemctl` and taking a physical backup, as well as adjusting settings before issuing the `systemctl start` command.

## MySQL Logical Upgrade to a New Host

The process for a logical upgrade of a MySQL server is quite straightforward and entails dumping the server with a tool such as `mysqldump` and then restoring it. In this article, I use `mysqldump`, but it's worth noting that you can get a speed boost from tools such as `mysdumper` that use parallel backup and loading. Although not an officially supported

```
'a' on (('s','address_id' = 'a','address_id')) join 'city' on (('a','city_id' = 'city','city_id')) join 'country' on (('city','country_id' = 'country','country_id')) */;
/*!50001 SET character_set_client = @saved_cs_client */;
/*!50001 SET character_set_results = @saved_cs_results */;
/*!50001 SET collation_connection = @saved_col_connection */;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2024-03-13 15:39:25
(base) → very_important_server
[0] 0:zsh* "forgeofdata" 11:39 13-Mar-24
```

Figure 3: Inspecting the output of a `mysqldump` backup.

approach, some have been able to upgrade directly from much older versions of MySQL directly to 8.0 in this way. If you choose this path, definitely test it thoroughly ahead of time.

For the sake of this example, assume you are using two instances – VMs, physical machines, or containers, it doesn't matter. One server will be the pre-existing MySQL 5.7 server, and the other will be your new MySQL 8.0 server. Technically, you can do this process on the same machine, but it's best to use separate instances if possible to make handling difficulties easier, as you will see.

Typically, you first take a backup from the old server by running the command

```
mysqldump -u root -p --all-databases | \
gzip > backup.sql
sudo systemctl mysql-server stop
```

Strictly speaking, the second command is not necessary but can be helpful to ensure no traffic is incorrectly routed to the old machine. Next, install a fresh copy of MySQL server 8.0, transfer the backup.sql file you made earlier, then run the commands

```
sudo systemctl mysql-server start
gunzip < backup.sql.gz | mysql -u root -p
```

If all goes well, this backup will be restored seamlessly. If not, for smaller databases, you can manually edit offending statements by running

```
gunzip backup.sql.gz
```

and then editing the file with a text editor. For larger databases, you can simply restart the original MySQL server on the original host, adjust or remove any offending features, and then re-dump and import – repeating the process as necessary until the restore succeeds.

In many cases, however, this procedure will go smoothly, and you can then switch traffic to your newly minted MySQL 8.0 server.

## Replication Topologies

In this article, I've discussed upgrading a server from MySQL 5.7 to 8.0. Of course, many organizations have long outgrown a single server and use a cluster or clusters of MySQL servers.

You can employ a similar process to handle any upgrade of any MySQL topology, in which you test applications, test the database, backup, upgrade, backup, and then resume processing transactions. Of course, this process becomes more complex to manage with more machines. Note that a MySQL 8.0 machine can replicate from a 5.7 machine, but not vice versa. You can, for example, add a new 8.0 replica to an existing 5.7 source to verify correctness – perhaps through the use of the pt-upgrade tool mentioned earlier.

Similarly, you can upgrade your 5.7 replicas one at a time before taking down your 5.7 source and upgrading that. In this way, you can reduce downtime because your source – and therefore your cluster – only needs to be down for a relatively brief period of time.

Finally, note that cloud products like Amazon Relational Database Service (RDS) and Google Cloud SQL typically have their own routines for handling the 5.7 to 8.0 upgrade; under the hood, it will likely be similar to the approaches discussed here, but the interface and procedures used will vary, so you should follow the procedures outlined in the vendor documentation.

## Conclusion

Generally, the changes implemented in MySQL 8.0 are for the best (e.g., utf8mb4 is a better default than latin1); nevertheless, they could pose significant obstacles for the unwary. If you aren't prepared for an upgrade now, the other option is to team up with one of the vendors offering extended MySQL 5.7 support beyond the official EOL date.

Fortunately, although it might be rather difficult to avoid death and taxes, an ill-planned MySQL upgrade is one thing you can certainly avoid. With a little foresight, you can confidently and successfully pull off your upgrade. ■

---

*This article was made possible by support from Percona LLC, through Linux New Media's Topic Subsidy Program ([https://www.linuxnewmedia.com/Topic\\_Subsidy](https://www.linuxnewmedia.com/Topic_Subsidy)).*

---

### Author

David Berube is the president of Durable Programming LLC, a boutique software development firm near Boston. He is the author of *Practical Ruby Gems* and *Practical Reporting with Ruby and Rails* and loves speaking about MySQL, PostgreSQL, and whatever exciting open source database comes out next.





Simple, small-scale Kubernetes distributions for the edge

# Right-Sized

We look at three scaled-down, compact Kubernetes distributions for operation on edge devices or in small branch office environments. By Andreas Stolzenberger

**Production Kubernetes** clusters use several physical servers, whether the Kubernetes nodes run directly on the hardware or use a virtualization layer – although not actually needed today. Many articles published about Kubernetes rely on single-node setups for practical examples because it is what application developers primarily use. At the same time, the number of scenarios in which single-node setups also make sense in practical use is increasing. More and more users are equipping edge devices with a simple Kubernetes environment.

On one hand, you can use a central cluster management system such as Open Cluster Manager [1] to manage these devices. On the other hand, you only need to develop and test your applications for a single platform: Kubernetes. A practical example includes merchandise management

systems. The components for warehousing, invoicing, and ordering run on the central Kubernetes clusters in the data center, whereas small edge servers with the point-of-sale (POS) application in a Kubernetes container are fine for the in-store POS systems. More powerful Kubernetes platforms (e.g., Rancher (SUSE) or OpenShift (Red Hat)) are not easy to set up on a single node. Although technically feasible, it makes little sense because full-fledged platforms run several dozen containers themselves. For this reason, various manufacturers offer lightweight Kubernetes distributions that only need a few containers for edge operation.

In this article, I look at three of these distributions: K3s (SUSE), MicroShift (Red Hat), and MicroK8s (Canonical). Of course, you will find other lean distributions, such as Kubernetes in

Docker (KinD), Minikube, and k3d, but I do not look at those options here because they are primarily intended for use on developer desktops.

## K3s

K3s [2] is the smaller sibling of the Rancher Kubernetes Engine (RKE), which became part of SUSE in December 2020. The distribution has been around since 2019 and can be operated with minimal resources. According to the manufacturer, K3s even runs on machines with only one CPU core and 512MB of RAM; the minimalist K3s setup itself only uses 250MB. As one of the radical cost-cutting measures, K3s dispenses with the I/O-intensive etcd database for configuration storage and uses SQLite instead, which is fine for single-node operation on an edge device.

Lead image © grafiner, 123RF.com

K3s also supports multinode operation. The distribution distinguishes between a K3s server, which runs Kubernetes management pods, and a K3s agent, which only runs application pods. This ability turns out to be very practical in edge use. If a single edge device can no longer handle the workloads on its own, you simply add a second agent node. That said, K3s also lets you operate several K3s servers for control plane redundancy, but with etcd in this case.

K3s thus has a flexibility that other small Kubernetes distributions lack: You can start with a single node and expand the Kubernetes environment as required during operation. K3s can convert an existing single-node setup with SQLite to etcd and expand the control plane to the usual three-node setup.

Further cost-saving measures can be identified in the storage, network, and proxy realms. K3s relies on the Traefik reverse proxy for IP routing instead of the standard but more resource-intensive NGINX, so you might need to change the existing deployment, stateful sets, or both configurations of your applications. Ingress routers are often defined by NGINX and not by

Traefik. Flannel, the simple overlay network with VXLAN, is used for the virtual pod networks. Other container network interface (CNI) drivers are available as options.

By default, K3s uses the simple Hostpath driver as the storage provider. Although it only supports filesystem storage and does not offer redundancy, it works fine for edge operation. Hostpath has no special requirements in terms of the node's disk or logical volume manager (LVM) setup and is also frugal in its use of resources. As with the network, K3s is flexible when it comes to storage. If Hostpath is not up to the task, an arbitrary other storage provider (Longhorn, Rook) can be retrofitted.

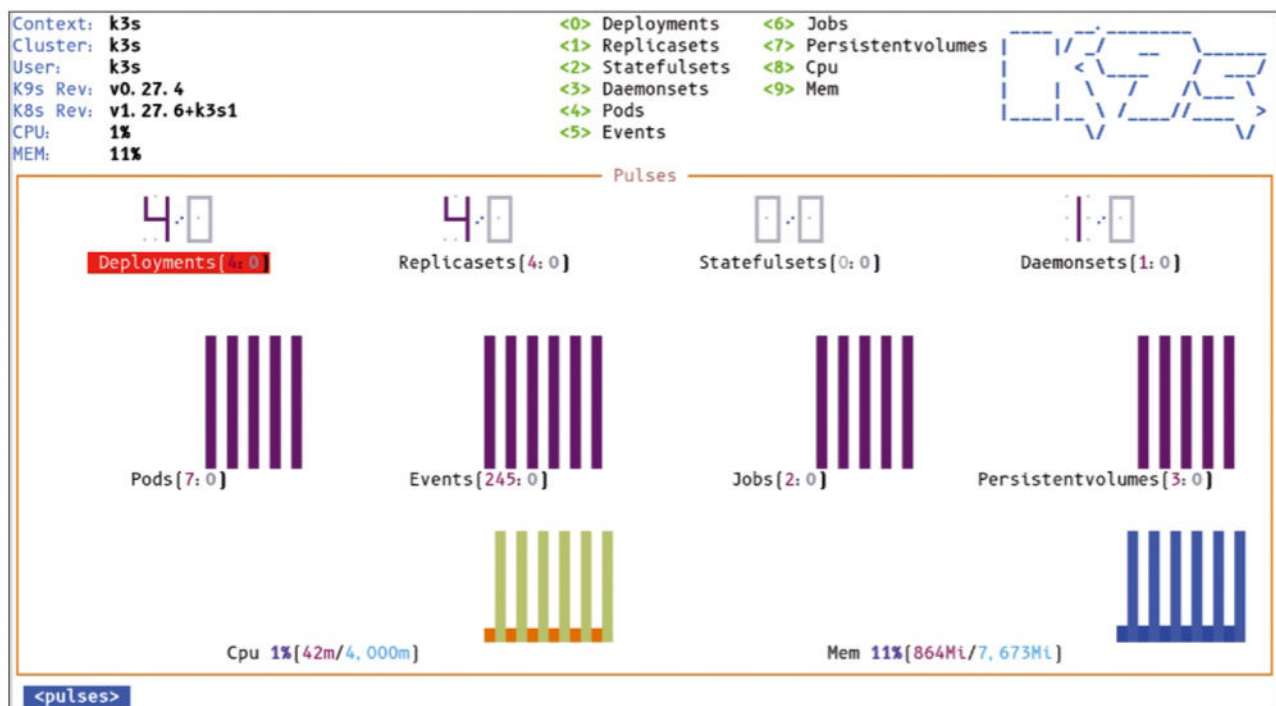
Installing K3s could not be easier. Almost all common Linux distributions are supported, including SUSE Linux Enterprise Server (SLES), openSUSE, Red Hat Enterprise Linux (RHEL), Fedora, CentOS Stream, and Enterprise Linux clones (with active SELinux), as well as Ubuntu, Debian, and Raspberry Pi OS. In terms of CPUs, the small Kubernetes supports the usual suspects, 64-bit ARM and x86\_64 processors, and the distribution is even said to run on the s390x. Like most

Kubernetes distributions, K3s uses CRI-O as its container engine. Thanks to 64-bit ARM support, K3s also runs on NVIDIA developer boards (e.g., Jetson, AGX, Xavier, and Orin) with support for the GPU installed there. The installation script can be downloaded and executed directly. It installs the packages and repositories required for the setup and launches the local services and pods. In a single-node setup without any special configuration changes, you can have Kubernetes up and running on a computer within a few minutes (Figure 1). To manage the setup, you can use the command line or client admin tools such as k9s or OpenLens.

All told, K3s is an easy-to-install, resource-saving, and extremely flexible distribution. I like the option of expanding a single-node setup to include simple agents or subsequently upgrading the setup to a cluster.

## MicroShift (RHDE)

The lightweight edge version of Red Hat's OpenShift unofficially goes by the name "MicroShift" [3] because the trademark for this term (spelled "microSHIFT") is owned by



**Figure 1:** Running the basic K3s setup, including the active metric collector, required just seven pods (six active containers) in the test setup and used less than 1GB of RAM.

a company that manufactures bicycle gears. The Red Hat Device Edge (RHDE) project has existed since 2022 and is still in the development phase. You currently require a Red Hat account to install. MicroShift loads container images from the closed Red Hat Container Registry, requiring a pull secret to do so. A free developer account is all you need.

MicroShift packs the essential Kubernetes services into a single systemd service on the edge host. This package also includes etcd as the config store. According to the manufacturer, 1GB of RAM and one CPU core (ARM64 or x86\_64) are all you need. MicroShift itself requires 500MB of RAM (more than you need for K3s), because NGINX for the reverse proxy, etcd, Open vSwitch (OVN instead of Flannel networking), and TopoLVM modules are used (Figure 2).

The installation on RHEL 9 relies on an RPM package and the DNF package manager. To install, you first need to add two repositories. Alternatively, MicroShift can be bundled into an OSTree image by the image builder. This rollout path is preferred for Kubernetes edge computing distributions, as well. The setup requires a special layout of the connected disk. Because MicroShift relies on TopoLVM as the storage driver, you must have a volume group named *RHEL* on the system with free space for logical volumes. When initially setting up the RHEL system, you must ensure that the “root” LV leaves you with sufficient free space.

MicroShift also uses the CRI-O container runtime, which the setup routine installs automatically. When first launched, MicroShift needs a few minutes to start the associated pods and services. TopoLVM in particular as a storage driver tends to take a little more time.

In addition to the regular Kubernetes APIs, MicroShift provides a few OpenShift extensions for security and routing. OpenShift routing as an alternative to Ingress routing is particularly interesting for users who use OpenShift on other clusters, which means you do not have to adapt existing deployments and stateful sets, at least in terms of routes. Multinode operation is planned for future versions but has not yet been implemented.

The lightweight edge version of OpenShift already offers solid functions but cannot keep pace with the flexibility of K3s as of this writing – particularly with optional multi-node operation. What I liked about MicroShift was its integration with the image builder for OSTree at the edge and that OpenShift routing is included. As an open source project, however, MicroShift is currently still too heavily dependent on commercial products such as RHEL and the closed Red Hat Registry. One hopes a completely open source project will appear in the future that will then also run on free systems such as Fedora without a Quay account. The current upstream documentation also leaves a lot to be desired: It describes how to set up version 4.8

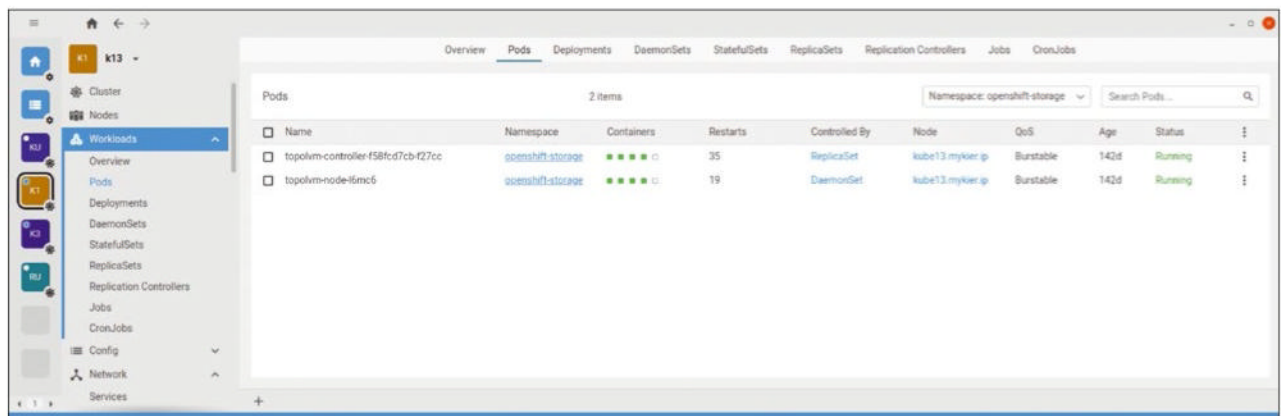
from April 2022 and not the current 4.14 version.

Whether the MicroShift project’s switch from the Hostpath storage driver to TopoLVM was a good idea remains to be seen. Although TopoLVM supports functions such as persistent volume (PV) snapshots, it has higher resource requirements (no fewer than eight containers), more time for PV provisioning, and a more complex disk configuration.

## MicroK8S

MicroK8s [4] is Canonical’s lightweight version of Kubernetes. Here, too, the manufacturer specifies the minimum RAM requirement of the distribution at around 500MB. The basic installation does not include an ingress router such as Traefik or NGINX, nor does it install a default storage driver. Canonical groups these functions in add-ons, which can be installed on top with the `microk8s` command. In this way, users can pick and choose what they need for their setups. MicroK8s also does away with etcd as the config store in the basic version. A proprietary distributed in-memory variant of SQLite, Dqlite, is used instead.

Unlike most other Kubernetes distributions, MicroK8s does not rely on the usual CRI-O container engine preferred by the Kubernetes project. Instead, Canonical, like Docker, embeds containerd directly. To make matters worse, the MicroK8s setup is not supported by the regular (Ubuntu/



**Figure 2:** Whereas K3s and MicroK8s use Hostpath as the storage driver with just one container, MicroShift relies on TopoLVM, which grabs eight containers.



Debian) Apt package manager but uses Canonical's Snap packaging system. Therefore, the required binaries and configuration files are not stored in the default Linux directories (e.g., /etc/, /var/) but somewhere below /snap (/microk8s/xxx/, /bin/), making debugging and troubleshooting more difficult.

Installations with CRI-O, for example, provide the `crictl` tool to retrieve information from running containers and images. A similar function for containerd is supposed to be provided by `ctx` but does not work well because of Canonical's weird Snap setup. As a result, MicroK8s does not work on systems with SELinux – only with Canonical's AppArmor. Calico is used as the network driver instead of Flannel. MicroK8s can also work with other CNIs, relying on add-ons, such as Kube-OVN (Open vSwitch), to do so.

The installation on an Ubuntu 22.04 LTS server is very simple. You can add

the MicroK8s Snap during the operating system installation. The `microk8s` command-line tool provides information about the current setup and also manages the desired add-ons. To add nodes, you use `microk8s-ctl`. The tool generates an individual token for each new node; a new host that you set up can use the token to join the cluster later. The running system is managed with `microk8s kubectl` or with the `kubectl` tool on a workstation. MicroK8s provides the Kubernetes dashboard as an add-on (Figure 3).

## Home Brew

Canonical's mini-Kubernetes setup also gives you a great deal of freedom at setup. The distribution will run on a single node without network ingress but also on a cluster network with etcd and OVN. The distribution's add-on strategy is particularly appealing to Kubernetes newcomers. Tools

that admins would otherwise have to set up manually as YAML files with deployments and role-based access control (RBAC) configurations are set up with a single `microk8s enable` command.

The downside is Canonical's self-built components and constructs such as Dqlite, K8s on containerd, and the confusing Snap setup. Where other manufacturers stick to established standards and tools supported by a large community (projects such as CRI-O, etcd, and SQLite have at least 10 times as many developers and commits compared with the Canonical tools), Canonical has gone for a do-it-yourself solution.

Over the years, Canonical has repeatedly tried to grab users' attention with alternative solutions to current standard tools – usually with little success. While OpenStack was establishing itself as a scale-out virtual

# 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](https://bit.ly/HPC-ADMIN-Update)

Linux Update: [bit.ly/Linux-Update](https://bit.ly/Linux-Update)

machine (VM) architecture, Canonical was backing the largely unknown Eucalyptus. Ubuntu chose LXC/LXD instead of the more popular Docker and the Unity desktop instead of Gnome – neither of which helped Canonical make more friends. You will want to consider carefully whether to invest resources and learning overhead in tools and architectures that might disappear in the long term.

## Conclusions

All three small distributions in this brief overview offer a relatively simple and compact Kubernetes distribution for operation on an edge device or in a small branch office environment. MicroShift provides very good edge integration with the OSTree image builder but currently only runs as a single-node setup. At its current stage of development, the distribution is too closely tied to the commercial Red Hat products and services. As an

open source project, the tool should not have to rely on closed registries. MicroK8s from Canonical makes it very easy for users to get started with Kubernetes and scales from a single node with a reduced feature set to a cluster, if required. The various services are bundled as add-ons and can be set up with a single command. The lack of SELinux support is a negative factor. Canonical's in-house tools such as Dqlite and the confusing Snap setup are also unlikely to impress many users.

K3s is a flexible distribution that also scales from a simple edge setup to a small cluster. The environment is very easy to set up, runs on various distributions, and supports SELinux. K3s also banks on well-maintained standard projects such as CRI-O and SQLite across the board. The migration option, which lets you switch from SQLite to etcd during operation if you want to upgrade your single-node setup to a cluster, is also appealing. ■

### Info

- [1] "A multicluster management tool for Kubernetes" by Andreas Stolzenberger, *ADMIN*, issue 76, 2023, [\[https://www.admin-magazine.com/Archive/2023/76/A-multicluster-management-tool-for-Kubernetes/\]](https://www.admin-magazine.com/Archive/2023/76/A-multicluster-management-tool-for-Kubernetes/)
- [2] K3s: [\[https://k3s.io\]](https://k3s.io)
- [3] MicroShift: [\[https://cloud.redhat.com/blog/meet-red-hat-device-edge-with-microshift\]](https://cloud.redhat.com/blog/meet-red-hat-device-edge-with-microshift)
- [4] MicroK8s: [\[https://microk8s.io\]](https://microk8s.io)

### The Author

**Andreas Stolzenberger** worked as an IT magazine editor for 17 years. He was the deputy editor in chief of the German *Network Computing* magazine from 2000 to 2010. After that, he worked as a solution engineer at Dell and VMware. In 2012 Andreas moved to Red Hat. There, he currently works as principal solution architect in the Technical Partner Development department.

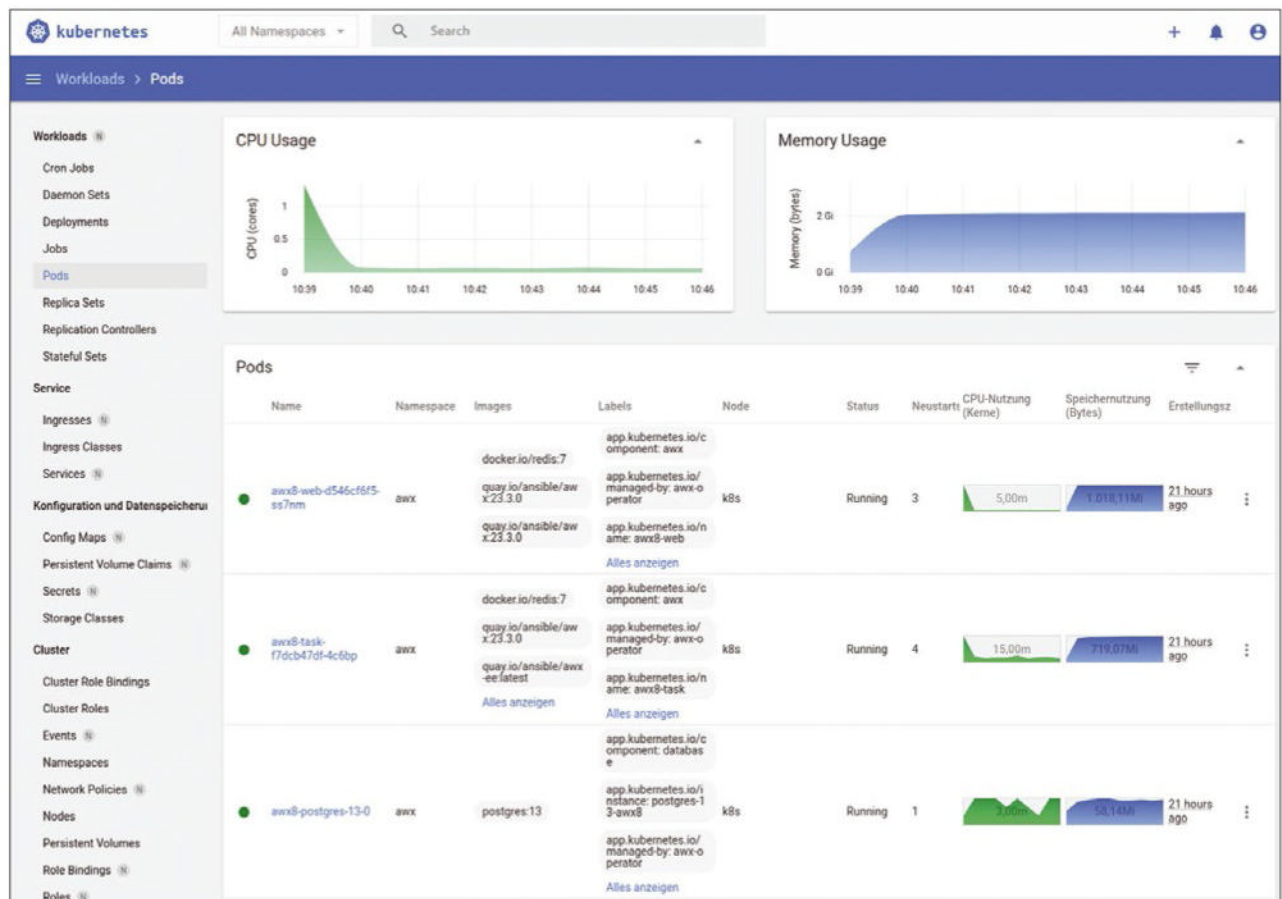


Figure 3: MicroK8s installs the Kubernetes dashboard as an admin user interface with a single command.





# PyCon US · 24

May 15th - 23rd Pittsburgh, PA



PyCon US is the largest annual gathering for the community that uses and develops the open-source Python programming language. We'll have talks, tutorials, social events, and more for all levels and uses of Python. We'd love for you to join us!



Scan me!

REGISTER AT  
[HTTPS://US.PYCON.ORG/2024](https://us.pycon.org/2024)



PYCON US 2024 IS A PRODUCTION OF THE PYTHON SOFTWARE FOUNDATION

BY THE COMMUNITY

FOR THE COMMUNITY



## Provisioning resources with an Azure-specific language

# Flex



Microsoft Bicep is a fairly new language for creating Azure resources in the cloud. We talk about why Bicep was developed, what the new language can do, and the advantages it offers admins. By Nico Thieme

**Microsoft Bicep is a relatively** new declarative language for defining Azure resources, with a user-friendly syntax that makes it far easier to create storage accounts, virtual machines, and resource groups. The aim of this article is not to cover all of Bicep's language elements – it has far too many to even attempt. Instead, I provide an overview of what the language can do in general and where the biggest differences lie compared with conventional JSON ARM templates. The advantages of Microsoft Bicep over

ARM templates include easier readability, simpler syntax, the ability to work with conditions, and more convenient handling of modules.

### Exclusively for Azure

ARM templates have been available for a very long time to help admins create resources in Azure in a declarative way. These templates are built according to the JSON standard, a universal format for transferring and storing structured data. It was not developed specifically

for Azure and can quickly become confusing as the data becomes complex.

Microsoft Bicep sets out to remedy precisely this shortcoming. The language was developed exclusively for Azure. One aim was to avoid the complexity of ARM templates and simplify the process of defining resources. Bicep files can generally manage with considerably fewer lines of code than ARM templates – and with fewer parentheses. On the other hand, the language can only be used for the Azure cloud.

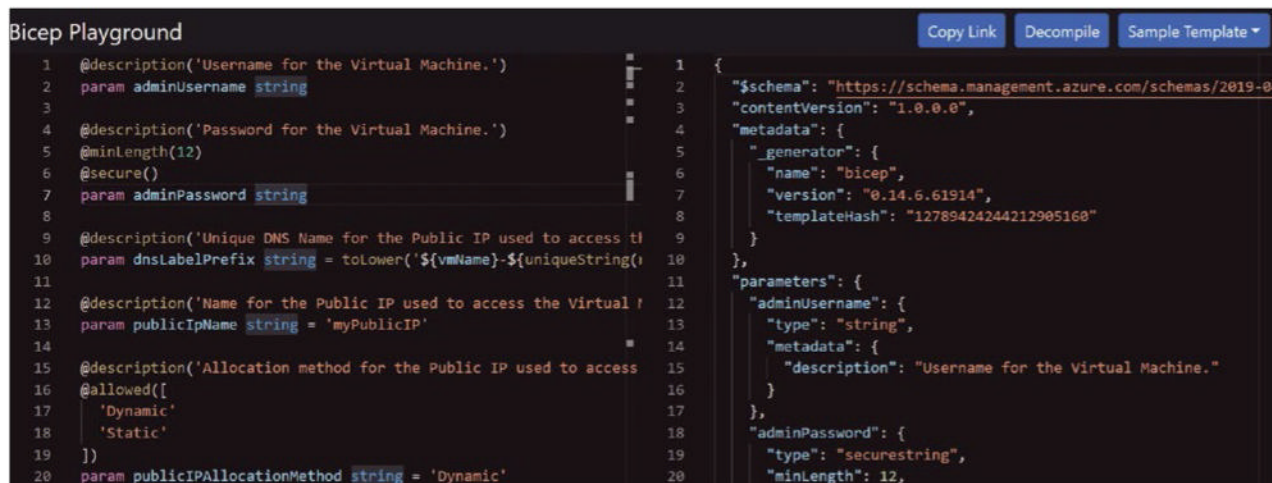


Figure 1: The Bicep Playground with a Bicep file on the left and an ARM file on the right.

## Script Conversion

Azure does not execute Bicep code directly but converts it into traditional ARM templates. At the end of the day, Bicep can only do what ARM templates can do. Of course, ARM files can also be converted to Bicep, for which Microsoft provides a sample application [1] and instructions [2]. Usefully, it contains many templates, so you can start using Bicep straight away.

Figure 1 shows a `main.bicep` template on the left and the matching ARM file on the right. Both are comparatively simple, which makes them useful as examples. If you look at the Bicep file, you will immediately notice patterns similar to those in familiar languages (e.g., parameters declared with `param` and variables with `var`).

Parameters can be assigned default values, and you can specify the values the parameter can assume – all with comparatively little code. The following snippet declares a `publicIPAllocationMethod` parameter with the default value `Dynamic`; the other value allowed is `Static`:

```
@description(
  'Allocation method for the Public IP
  used to access the Virtual Machine.')
@allowed([
  'Dynamic'
  'Static'
])
param publicIPAllocationMethod
  string = 'Dynamic'
```

The corresponding code for the ARM template is far more complex and longer (Listing 1). Declaring resources (e.g., a storage account in this example) in Bicep is also simpler:

```
resource stg 'Microsoft.Storage/2
storageAccounts@2021-08-01' = {
  name: storageAccountName
  location: location
  sku: {
    name: 'Standard_LRS'
  }
  kind: 'Storage'
}
```

The convention is for the resource keyword to be followed by a symbolic name, which is not the name of the resource in Azure; instead, you use the `name:` parameter. The symbolic name follows the naming conventions of the resources in Azure and is usually a parameter. The symbolic name of the resources is followed by the resource type; the API version can also be specified with `@`.

Visual Studio Code comes with Bicep already integrated, or you can use IntelliSense or simply rely on a visual preview of the resources defined in Bicep. Bicep templates can be run on the client side by PowerShell or the Azure command-line interface (CLI), which presupposes a comparatively complex installation procedure – detailed instructions can be found online [3].

Luckily, you can save yourself all of this trouble and run the script in the Azure Cloud Shell, although you will need to upload the script each time:

```
Connect-AzAccount
# Only if not Cloud Shell
New-AzResourceGroupDeployment
-ResourceGroupName projectx1000
-TemplateFile "C:\Scripts\bicep\
storage.bicep"
# the path is different in the cloud shell
```

This example shows a call with PowerShell in Windows.

## Other Language Constructs

Working with loops and conditions is far easier in Bicep than in JSON. Even if you only have basic programming skills, you will be able to find your way around. Assume you want

to create two storage accounts, which obviously need unique names (Listing 2). The word `for` iterates over the `storageCount` parameter, creating the two storage accounts in the process. Besides the parameters being distinguished by number, they are also assigned unique names with the `uniqueString` function by generating a hash from the resource group ID. Bicep supports conditions, including `if`, which means that decisions can be made quickly and easily and is particularly interesting if you want to decide whether or not to provide resources as a function of some parameters.

The example in Listing 3 assumes a staging environment and a production environment. If the script is executed in the staging environment, the resource is not created; however, you do want it to be created in the production environment. If you pass arguments to a Bicep script when it is called, you can use the same script to create different resources in different environments.

To add output, use

```
output <name> <data-type> = <value>
```

### Listing 1: ARM `publicIPAllocationMethod`

```
01 "publicIPAllocationMethod": {
02   "type": "string",
03   "defaultValue": "Dynamic",
04   "allowedValues": [
05     "Dynamic",
06     "Static"
07   ],
08   "metadata": {
09     "description": "Allocation method for the Public
10       IP used to access the Virtual Machine."
11   },
```

### Listing 2: Creating a Storage Account

```
01 param location string = resourceGroup().location
02 param storageCount int = 2
03 resource storageAcct 'Microsoft.Storage/storageAccounts@2021-06-01' = [for i in range(0, storageCount):
04   {name: '${i}storage${uniqueString(resourceGroup().id)}'
05     location: location
06     sku: {
07       name: 'Standard_LRS'
08     }
09     kind: 'Storage'
10   }]
```

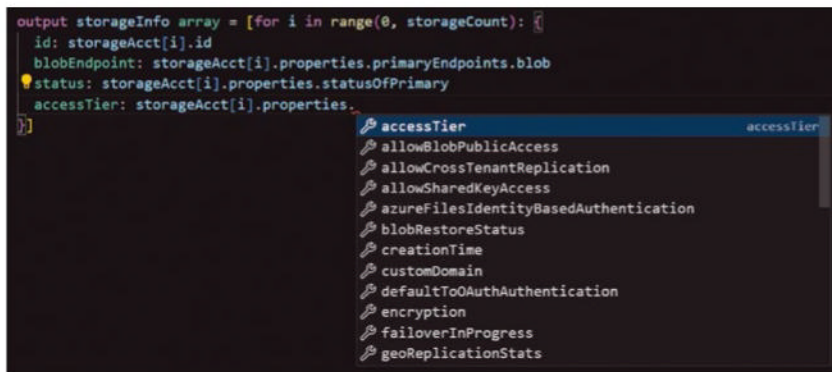


Figure 2: Bicep output with parameters.

In other words, each item of output needs a data type in addition to the output value. The ability to access all the values of a previously created resource is useful. Figure 2 shows an example of the kind of output you can generate when writing in Visual Studio Code after creating two storage accounts with the available parameters. The output itself offers many options, which you can discover in the documentation [4].

## Working with Modules

Another advantage of Bicep is the simple use of modules. Other languages use include files, which is

possible with ARM templates, as well, although the process is far more complex. To outsource the declaration of the memory account to a module in Bicep and call it up later, use:

```
module stgModule '
  './storageAccount.bicep' = {
    name: 'storageDeploy'
    params: {
      storagePrefix: 'examplestg1'
    }
  }
```

The module also has a symbolic name, which is used later to access the module and its output. The name is followed by the name and path of the module file, followed by parameters, which are optional; you can also specify all the parameters in the module file, which itself is a Bicep script without any further declarations.

To generate output with the values from the module, simply reference the module name and add outputs; then, add the property you want to retrieve. Note that the output is of the object type:

```
output storageEndpoint object =
  stgModule.outputs.storageEndpoint
```

Many organizations already have a large number of ARM files defining recurring

resources. It would make little sense to convert them all to Bicep and is not necessary because Bicep can use both its own and ARM templates as modules. In this case, simply include the ARM file when connecting the module. One possible path to a smooth transition would be to leave the existing ARM files as they are and simply write new code in Bicep.

However, a few reasons might lead you to avoid working with Bicep at all. For example, if you have only recently familiarized yourself with ARM, it might be difficult to learn a new language straight away. Although Bicep is more convenient and more lightweight than ARM templates, it ultimately does the same thing. If you use a cross-cloud framework for infrastructure as code and this framework only outputs an ARM, not Bicep, script you would then have to convert the files to Bicep. The question is whether it is worth it.

## Conclusions

The Bicep language was developed specifically for provisioning resources in Azure and has far simpler syntax than ARM templates. Training is straightforward, the probability of error is low, and it is the tool of choice for newcomers; however, Bicep also offers many useful features for old hands.

### Listing 3: Parameters

```
01 param storageAccountName string
02 param location string = resourceGroup().location
03 @allowed([
04   'stage'
05   'prod'
06 ])
07 param stageOrProd string = 'stage'
08 resource sa 'Microsoft.Storage/storageAccounts@2019-06-01' =
    if (stageOrProd == 'prod') {
09   name: storageAccountName
10   location: location
11   sku: {
12     name: 'Standard_LRS'
13     tier: 'Standard'
14   }
15   kind: 'StorageV2'
16   properties: {
17     accessTier: 'Hot'
18   }
19 }
```

### Info

- [1] Bicep Playground: <https://azure.github.io/bicep/>
- [2] Decompiling ARM template JSON to Bicep: <https://learn.microsoft.com/en-us/azure/azure-resource-manager/bicep/decompile>
- [3] Manual install of Bicep CLI: <https://learn.microsoft.com/en-us/azure/azure-resource-manager/bicep/install#install-manually>
- [4] Output in Bicep: <https://learn.microsoft.com/en-us/azure/azure-resource-manager/bicep/outputs>



# Hone Your Skills – with – Special Issues!

Get to know Shell, LibreOffice, Linux, and more from our Special Issues 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



background image © roystudio, 123RF.com



Check out the full library!  
[shop.linuxnewmedia.com](https://shop.linuxnewmedia.com)



A self-hosted server and site manager for WordPress

# Serving Up Words

WPcloudDeploy helps reduce hosting costs for your fleet of WordPress cloud servers by managing dozens or even hundreds of servers and sites with WordPress-specific smarts. By Nigel Bahadur

## A recent trend in the WordPress

realm is WordPress-specific control panels (see the “Overview” box). One of the advantages of these panels is the sheer depth of WordPress idiosyncrasies that are baked in and hidden

### Overview

By most measures, WordPress is the largest content management system (CMS) in use by far, with a more than 40 percent market share [1]. The next closest CMS is Wix with only a 3.6 percent market share. Although you can argue about actual percentages on the basis of who's doing the reporting, almost every report has WordPress market share at more than 10 times the next closest competitor.

One of the consequences of this lopsided lead has been the rise of WordPress-specific services, starting with dedicated WordPress hosts such as WP Engine. Next, general hosting companies created WordPress-specific offerings with dedicated WordPress tools for management, backups, migration, and more. Although Plesk and cPanel hosting control panels are imbued with WordPress-specific functions, they are still general-purpose tools that aim to be all things to all people. Two of the first WordPress-specific control panels were GridPane and SpinupWP. If you wanted to host sites other than WordPress on these services, you could not do so. For them, it was WordPress all the time.

from your average WordPress users. Another advantage is that the panels connect to server providers such as DigitalOcean, AWS, and others. The panels themselves do not sell server resources; instead, they connect you to your accounts at one or more CloudServer providers and allow you to install your WordPress sites there. Users of these panels can choose server sizes and server load (how many sites, what types, etc. go on each server). With this approach the cost of running a WordPress site can be as low as \$1 per site when running multiple sites – even if you include the monthly cost of the control panel. These days the market for these WordPress-specific control panels has expanded significantly. GridPane and SpinupWP now compete with offerings such as Cloudways, RunCloud, Ploi, and, most recently, xCloud. However, all these panels are software-as-a-service (SaaS) products that require a monthly fee – which is where WPcloudDeploy enters the picture.

WPcloudDeploy is an open source, self-hosted WordPress-specific control panel built on WordPress itself. As far as I know, there is no self-hosted or open source WordPress control panel

with a graphical user interface (GUI) other than WPcloudDeploy. The closest is WordOps, which is an amazing WordPress tool but is command line only and has nowhere near the features of WPcloudDeploy.

## Features

WPcloudDeploy is available on GitHub [2]. The GitHub landing page includes extensive explanations, links to documentation, and a full human-readable development history. WPcloudDeploy includes all the features you might expect from a WordPress control panel:

- Support for deploying unlimited servers and sites
- Automatic deployment of SSL certificates by LetsEncrypt
- Site cloning
- Pushing sites between servers
- WordPress-specific cron options
- Support for WordPress-specific page and object caching (Redis and Memcached object caches)
- Team support
- White label support
- Cloudflare integration
- MicroCRM (customer relationship management software)
- Multiple PHP versions

- A basic REST API
- A ton more WordPress-specific and WordPress-adjacent functions

## Shortcomings

As you might expect, you will encounter some shortcomings. Probably the most significant for many readers is that the only Linux distribution supported by WPCloudDeploy is Ubuntu. When the control panel deploys a WordPress server, its scripts assume that the server is running on Ubuntu 20.04 or 22.04.

The other shortcoming is the user interface (UI). Because it is based on WordPress and tries to adhere to the WordPress standards, the UI is as dated as WordPress's admin dashboard. In this day and age where everything looks and feels beautiful, the WPCloudDeploy WordPress admin UI might feel a bit jarring. Finally, the free version of WPCloudDeploy only supports direct connections to DigitalOcean accounts. (Only the premium version supports connections to the other major cloud providers, including support for certain ARM-based servers.)

## Installation

To use WPCloudDeploy you need an existing WordPress site. I'm sure some of you are going, "What ...? You need a WordPress site to install a WordPress control panel?" Yes, it's a bit meta. But, it gets better – the site has to be running on a web server with beefed up time outs and other limits, as well as having SSH ports open, which rules out sites on the most common WordPress hosts. Instead, the best way to get started is to use the DigitalOcean starter image [3]. The current version of the image is 5.5, so one of the first things you'll end up doing is upgrading the WP-CloudDeploy plugin after the image is used. However, I'm getting ahead of myself.

The alternative to using the image is to deploy your own Ubuntu 22.04 server and then follow the extensive instructions online [4]. For the

purposes of this article, I'll be using the DigitalOcean image because it's the fastest way to get up and running. If this is the first time you're using a DigitalOcean account, please make sure you add a public key to your account – you'll need to specify this key when you create the droplet. WP-CloudDeploy does not support the use of passwords. You can add your public key by going to *Settings | Security* in your DigitalOcean dashboard. Once you've confirmed that you have a public key, navigate to the WP-CloudDeploy page in the DigitalOcean marketplace [3] and click on the *Create WPCloudDeploy Droplet* button. You should select a droplet with at least 2GB of memory. For most use cases you only need one or two CPUs, but make sure you select your *Public SSH Key* to be added to the droplet. After you have created the droplet, you should pause and create a DNS entry in your domain to point to the droplet's IP address. This step is needed because the new droplet will attempt to obtain an SSL certificate when you first login. For example, you can set up a subdomain such as *wpcd. <yourdomain> .com*, *wppanel. <yourdomain> .com*, or even *wp. <yourdomain> .com*. Of course, you can quickly purchase a new dedicated domain and use a top-level domain, as well.

When you log in to the droplet the first time (with your SSH key pair), you'll be prompted to enter the domain for your new WPCloudDeploy panel (just the domain name, without *www* or *http://*, e.g., *<yourdomain> .example.com*). Next, you'll be prompted to enter your email address and a couple of other details. The WPCloudDeploy site will then be created, and you'll be shown the end screen with your user ID and password. You can find more detailed instructions for creating the droplet at the WPCloudDeploy documentation site [5].

## First Login

Now that the droplet has been prepared, you can navigate to the site in your browser. The URL you need to

use for the login screen is the standard *wp-admin* URL. So, if your domain was *wpcd. <yourdomain> .com*, you need to navigate to *wpcd. <yourdomain> .com/wp-admin*.

The user ID and password given to you at the end of the installation process should get you into the dashboard. There, you'll see a prompt at the very top of the page with a green background. You can ignore that for now because you should upgrade the plugin to the very latest version before doing anything else.

## Upgrade

Download the latest release of the plugin ZIP to your local drive from GitHub [6]. Next, in your new WP-CloudDeploy site, navigate to *Plugins | Add New Plugin* and upload the plugin file. Choose the option to replace the existing plugin. Finally, you should run all the updates under the *Dashboard | Updates* screen. You might have to run three sets of updates: WordPress, Plugins, and Themes.

## DigitalOcean API Key

One of the items you'll need to connect WPCloudDeploy to your DigitalOcean account is an API Key, so in your DigitalOcean dashboard, navigate to the *API | Tokens* tab. To continue, click on the *Generate New Token* button on the upper right, enter a Token Name, set the expiration to *No expire*, check the *Write* option (very important!), and click the *Generate Token* button (Figure 1).

## First Time Install Wizard

Now you can use the installation wizard at the top of the site. On your first login to your new WPCloudDeploy site, click on the button in the setup notice at the top. You should now be at the very first screen in the wizard (Figure 2). As the screen suggests, you can skip it and just hit the *Continue* button. The reason you can skip this screen is because the DigitalOcean droplet setup process already



configured the necessary items for this step.

The next screen is where you get to choose your server provider. Because you only have one

choice – DigitalOcean – you can, again, just click the *Continue* button to proceed.

The third screen (Figure 3) is where you need to enter your DigitalOcean

token. One quirk on this screen is that you frequently need to enter the token twice. The first time you paste it in it will likely tell you that it's incorrect – just paste it in again and it will usually work the second time – perhaps just a quirk of the paste process. Click *Continue* to move to the next screen, which just tells you it will be creating SSH keys next. This screen might seem confusing because you already have keys in your DigitalOcean account. In fact, you used them to create the droplet, so why do you need to create new ones? The keys you create on this screen will be used in new droplets you create inside WPcloudDeploy, which helps separate the security of your main WPcloudDeploy droplet from those you create inside the dashboard. Security isolation is always a great thing. WPcloudDeploy does have an option to use your existing keys, but you can't set them up for use within the wizard; you would need to go directly to the WPcloudDeploy *Settings* tab. For now, click the *Continue* button to create a new key pair, which should take you to the final wizard screen with a link to the documentation and a single button, that takes you to the server list screen to create your first server.

## Creating Your First Server

The server list will be blank (Figure 4), so you should create your first server by clicking the *Deploy A New WordPress Server* button at the top of the screen. A straightforward questionnaire asks for your web server details, including your operating system, provider, region, size, script version, and name of server. For the server size, make sure you choose one with at least 2GB of RAM. Finally, click the *Deploy* button at the bottom. The form disappears, and a black output “console” takes over. If all goes well, between 10 and 20 minutes later you'll see the final output. The end of the server creation process is not complete until you click the *OK* dialog box button and you see the *Installation completed!* message at the

Figure 1: The screen in your DigitalOcean console to request an API token.

Figure 2: The first screen of the setup wizard. You can skip this step and just continue to the next screen.

Figure 3: Enter your DigitalOcean API token. You might have to do this twice.

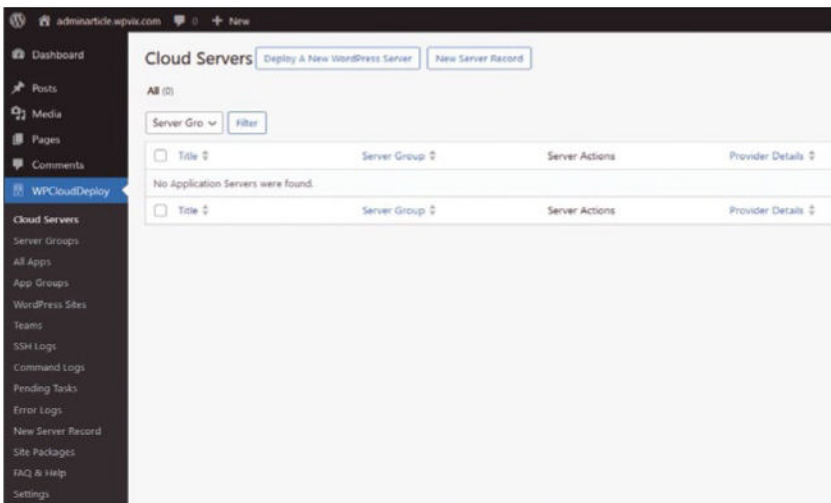


Figure 4: The blank server screen in the dashboard.

bottom of the Command Results output window. Click the *Close* button at the bottom of the window to get back to the server list, where you should see your first server (Figure 5). Servers created with WPCloudDeploy deploy either a LEMP stack (Linux, NGINX, MariaDB, PHP) or a LOMP stack (Linux, OpenLiteSpeed, MariaDB, PHP). For production use, I recommend the LEMP stack.

## Your First WordPress Site

Now that you have your first server, you can create your first site. Before you do, however, whatever domain or subdomain you're going to use should have a DNS entry updated to point to the server's IP address. The IP address for the server is shown in the Provider Details column of the server list, as well as on your DigitalOcean dashboard. To create your first site, click the *Install WordPress* button under Server Actions. You will only see one button because you only have one server. If you had multiple servers, you would see multiple buttons – one for each server row.

A simple form pops up that collects the essentials required to create a site (Figure 6). To create a WordPress site, click the *Install* button at the bottom of the form. As with the server, the form will disappear, and the black output console will take over in a separate pane. If all goes well, about five minutes later you'll see the final output.

The site is not complete until you click the *OK* dialog box button and you see the *WordPress has been set up* message at the bottom of the pane. Click the *Close* button beneath the console to get back to the server list, where you should see your first server

along with the site name in the Apps column. At this point you should be able to go to your browser and navigate to the site.

## Exploring the Site Options

Now that you have your site, what can you do? Most likely the first thing you would want to do is generate an SSL certificate so the site is served over HTTPS. To do this, navigate to the site details by going to *WPCloudDeploy | All Apps* and clicking on the hyperlink in the Title column for the site (you should only have one site in the app list). To generate the SSL certificate for the site, click on the *SSL* tab and then on the *Disabled* switch in the SSL section (Figure 7). After a couple of minutes of “thinking,” the option should turn green and the screen will refresh. Inside the details screen for the site are 25 tabs covering 95 percent of the things you would ever want to do

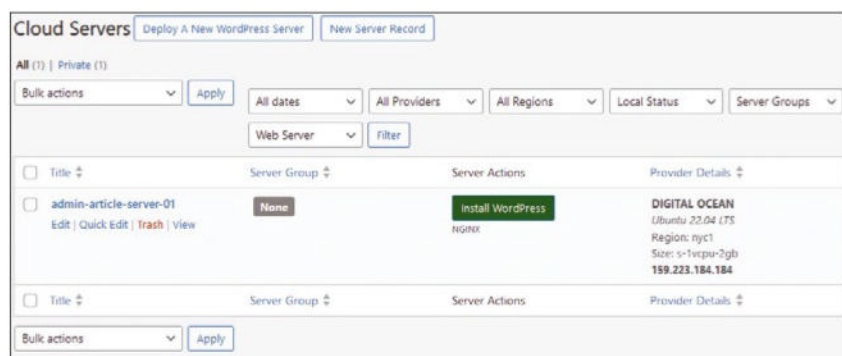


Figure 5: The server list showing your new server.

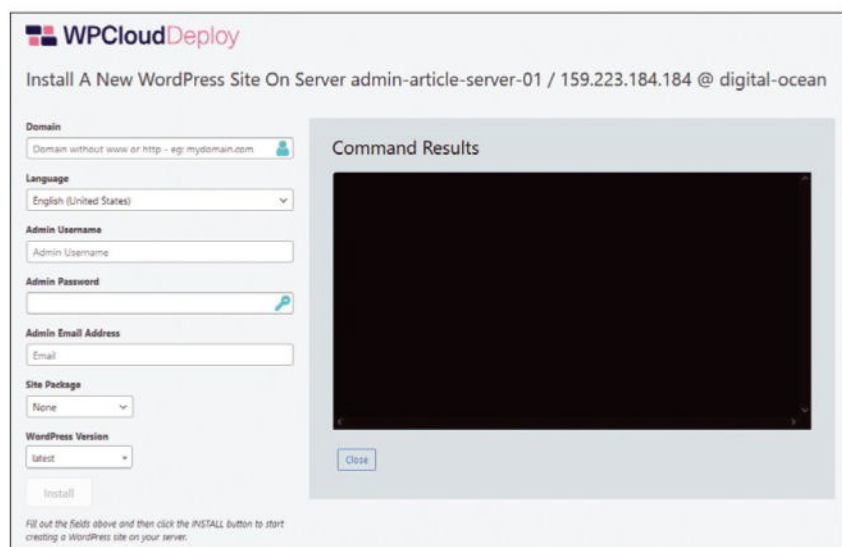
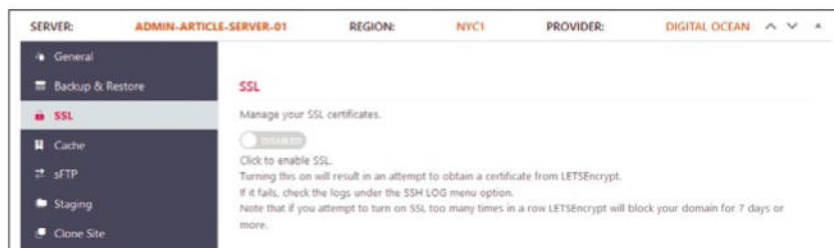


Figure 6: The site creation pop-up: You can leave the *Site Package* field set to *None*.



**Figure 7:** Click the toggle button to enable SSL. If successful, the toggle should turn green and the screen will refresh.

with a WordPress site, as far as managing it goes, including:

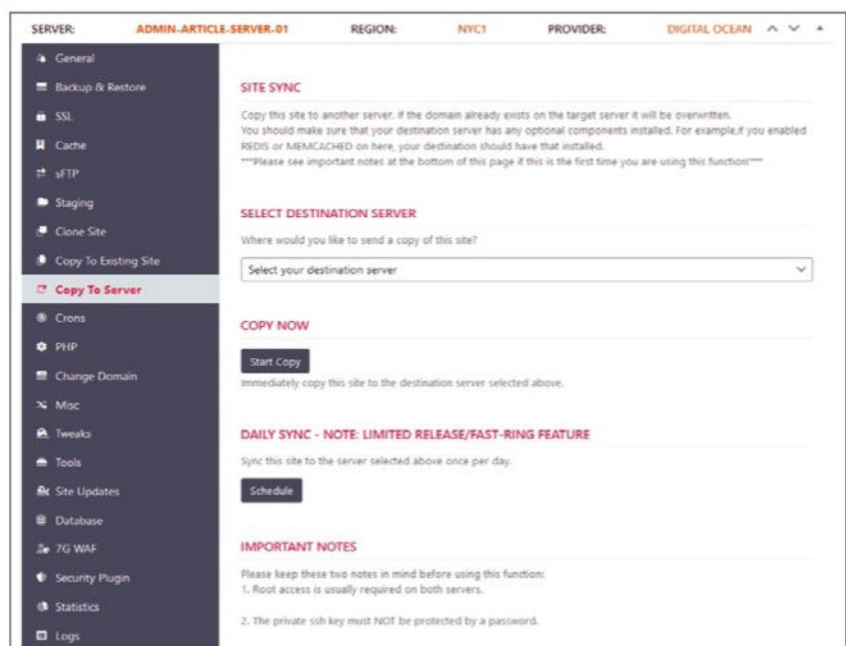
- Backing up and restoring to AWS Simple Storage Service (S3) or an S3-compatible endpoint
- Managing page and object caches
- Generating SFTP users
- Creating staging sites
- Cloning the site
- Copying over another site
- Pushing the site to a different server
- Managing Linux crons
- Changing the domain
- Managing PHP options (e.g., changing versions or setting file upload limits, memory limits, etc.),
- Installing phpMyAdmin to manage the database or a file manager to manage files inside the browser (as an alternative to SFTP)
- Viewing logs, managing wp-config
- Enabling/disabling web server bot rules (7G firewall)
- Setting up redirect rules

One of the things that make the WPCloudDeploy site seem cluttered is that it includes a lot of information “in your face,” so to speak. For example, **Figure 8** shows the option to push a site to a new server. In the UIs of other products, a lot of the options and information might be hidden under a hamburger (three stacked horizontal lines) or more vert (three vertical dots) menu option. In WPCloudDeploy, it’s up front, and for some folks it can be distracting, but you eventually get used to it.

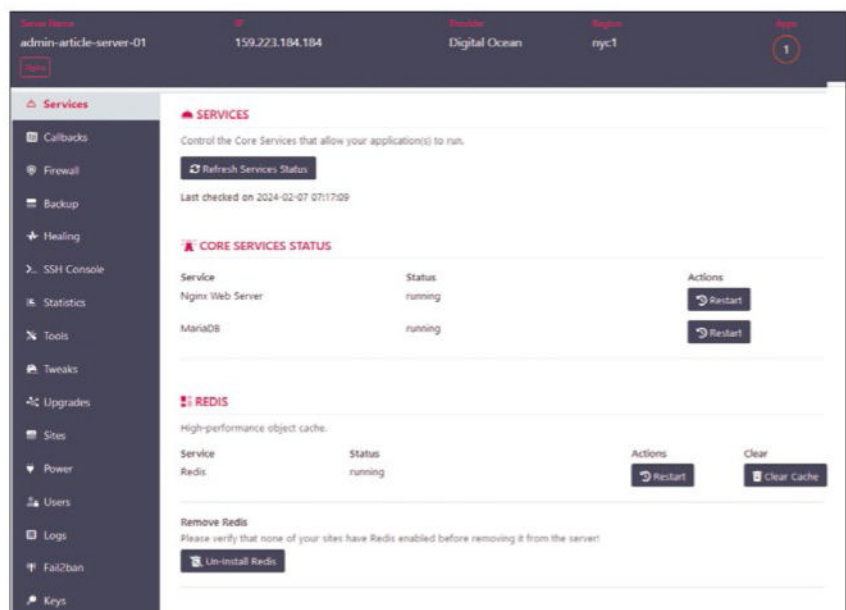
## Exploring Server Options

As with sites, you have extensive options for managing your servers inside WPCloudDeploy. Although at times you might need to SSH into a

server to get things done, those occasions should be few and far between.



**Figure 8:** The *Copy To Server* tab shows a screen with a lot of the informational text exposed instead of hidden behind pop-ups or icons.



**Figure 9:** A subset of the extensive options available for managing your WordPress server.

To view the management options for a server, click on the server title in the server list (**Figure 9**).

As is typical of WPCloudDeploy, most of the UI is text (which means it renders fast!). From this screen you can:

- Restart critical services such as MySQL, NGINX, etc.
- Set up server-wide backups for all sites (instead of setting them up for each site individually)
- Manage object cache servers
- Run Linux updates



- Manage Fail2Ban and the UFW firewall
- Restart and reboot (soft and hard reboot options)

You also can install and manage some non-critical tools such as Monitorix and GoAccess.

One nice option is integrated Monit [7]. If you're familiar with this tool, you'll love that it's an option because it helps you recover and restart services automatically when they're failing or using up too many resources. You'll find it under the aptly named *Healing* tab.

## Loosely Coupled Connection

WPCloudDeploy is loosely coupled to your servers. Changes you make in WPCloudDeploy will flow through to your server, but changes you make directly on your server outside the WPCloudDeploy screens will not necessarily be reflected inside the WPCloudDeploy panel. Moreover, even if your WPCloudDeploy panel instance is down, your sites will continue running on their servers. Contrast this with something like Plesk or cPanel where, in many instances, the panel is running on the same server as the WordPress sites. For those tools, when the panel is down, the sites can go down with it.

## Security and Teams

Security in WPCloudDeploy starts with that provided by WordPress. It makes extensive use of WordPress's Roles and Capabilities to lock down menu options. When first installed, WPCloudDeploy adds new pre-built roles you can assign to your internal or external users for whom you want to grant platform access. For example, you can create a user with a server-only role who can manage servers and sites without granting them access to the Team or Settings screen, or you can grant a user a site-only role for managing sites, without granting them access to the other screens.

As with standard WordPress, you can also customize the pre-built roles or

create your own. With teams you can fine-tune things even further and assign teams to servers and sites (Figure 10). Just about every tab can be locked down as needed, which nicely complements the WordPress roles that can lock down menu options, and you can assign each server and site to individual users. This feature makes use of WordPress custom post types. In WPCloudDeploy, most data about a server or site is stored in a custom post type. A WordPress custom post type record has an Author field that is generally used to indicate who created the record. In WPCloudDeploy, the Author field is used to determine who "owns" the site or server. This information is very useful when you're granting users access to sites and servers on the front end of the

WPCloudDeploy site. When a non-admin user logs in, only servers or sites they own will be accessible.

## WordPress Front End

When it comes to WordPress plugins and security, one of the first questions you might get is whether you can grant access to users for certain features on the front end of the site instead of giving them access to wp-admin. With WPCloudDeploy, you can do this; it even has a separate set of security policies for tabs when they are being rendered on the front end (Figure 11). The site typically takes on the fonts, sizing, and other elements of the site theme, which can be very different from how it renders in wp-admin.

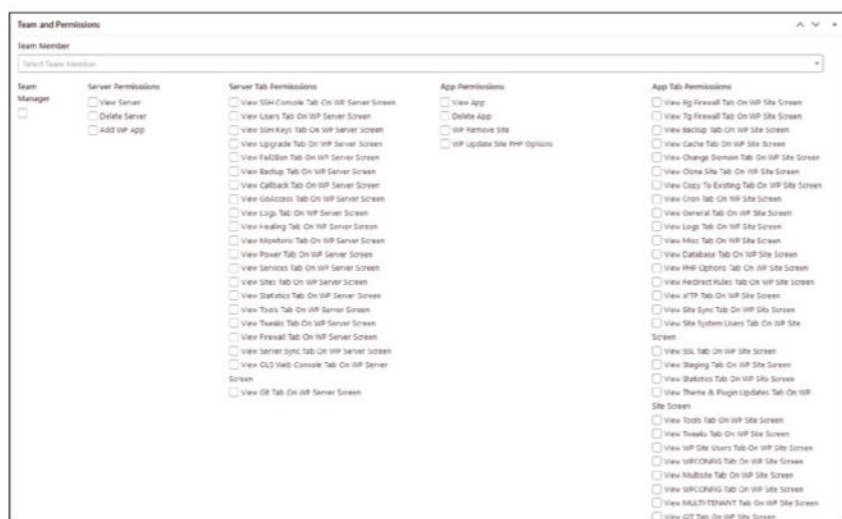


Figure 10: The extensive options available for locking down security on a team member.

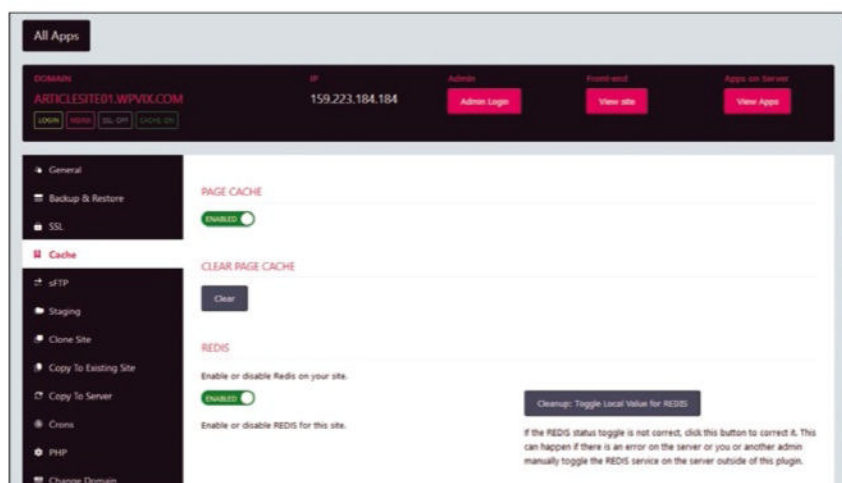


Figure 11: The site details on the front end of your WPCloudDeploy site take on the characteristics (e.g., fonts, page width, etc.) of the WordPress theme.

If you're a WordPress agency that would like to grant your users a management panel similar to the large hosting companies, this feature allows for that. You can lock down tabs by roles or, globally, for front-end users without affecting your wp-admin users. From a security perspective, most options have separate controls (Figure 12).

## White Label

As with anything related to the front end, you might not want the WPCloudDeploy brand to show up anywhere or you might want to customize the colors, logo, look and feel, and so on. WPCloudDeploy includes basic pre-built controls for white label scenarios with a UI for color and logo changes and the ability to render custom CSS on the plugin pages.

However, the real power in the tool is that it is open source and based on WordPress. These two elements grant you unprecedented control over what is rendered because you can customize everything in the code, WordPress style.

## WordPress Site Integration

The advantages of building your control panel inside a WordPress instance becomes apparent when you start thinking about things like

translations, customizations, and country-specific privacy rules. With SaaS services, your data is located wherever the service says it will be located and is rendered in whatever language and locale the SaaS owner deems important. With WPCloudDeploy you can place your panel and data wherever you like, and because it's a WordPress plugin, you can use all the WordPress translation tools to localize it for your region.

Because most of the data is stored inside custom post types (CPTs), you can use your favorite CPT plugins, such as Advanced Custom Fields, Admin Columns, and Meta Box to add fields to screens and lists, create new meta boxes that display the data differently, extract the data to display on custom front ends, and more.

Learning to customize is easy because you can use all your existing WordPress knowledge, and you only have to learn WPCloudDeploy-specific things. The product includes tons of WordPress-style hooks and filters, with several examples of extension plugins on GitHub, along with associated tutorials on the WPCloudDeploy site to show how to add new features without modifying the core WPCloudDeploy code. The availability of the hooks and filters make it possible to avoid modifying core code for many types of changes, which in turn

makes upgrading to new versions much easier.

## Third-Party Integrations

WPCloudDeploy includes a number of integrations, of which Cloudflare is one of the more important. With it, new sites can be assigned temporary subdomains automatically and made available immediately without the admin having to update DNS manually, which also makes SSL available without a lot of delays.

Other integrations include Logtivity, a logging service that prevents intruders from removing critical WordPress logs, and an image generation and comparison service used for automatic rollbacks after sites are updated. As you might expect, the premium product has many more integrations, especially for cloud server providers.

## Hooks and Bash Scripts

WPCloudDeploy includes hooks into custom Bash scripts at key operational points – in particular, just before creating a server is completed and just before a site is created. These hooks allow you to automate any customizations you might want for all your servers and sites with the Bash scripts you've come to know and love. It even automatically injects a user-defined global secret key into the Bash environment, which you can use to connect to your secrets manager (e.g., Doppler). With site packages, you can even have different scripts for different groups or types of sites.

## Auditing

Because WPCloudDeploy sends commands to your servers over SSH, you might want to see what it's sending and what's being returned over time. Three screens allow you to get this information: command log, SSH log, and error log screens. Command logs and SSH logs are similar: command logs are for long-running commands, such as creating servers and sites,

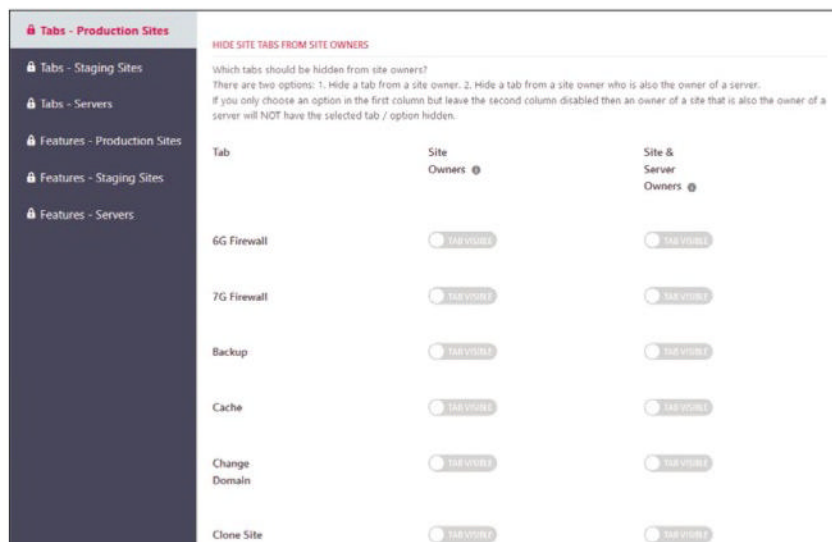


Figure 12: You can lock down tabs for users accessing site details on the front end of the site.

whereas SSH logs are for shorter requests expected to run within the PHP worker time out (**Figure 13**). Nothing goes into error logs until error log flags are turned on in the WPCloudDeploy settings, and the screen is generally not used except for debugging purposes. Another log screen for pending tasks keeps track of the progress of requests handled by WPCloudDeploy that take place in the background.

## Alerting

As an admin, you're going to want alerts for significant actions on your site. WPCloudDeploy includes a very flexible registration and alerting function that starts by decoupling the recording of events with the alerts for those events and displays them in the Notification Logs screen (**Figure 14**).

You can set up alerts for different events and send each to a different subset of users as necessary. Alerting targets can include email addresses, as well as Slack, Zapier, and generic webhooks. Alerts can be filtered by server, site, type (e.g., notice vs. warning), and "reference," which includes things like "backup" and "power." For example, you can set up an alert for backups that go to one admin and another for power events that go to another admin. If you're providing front-end access, your end users can set up their own alerts there, as well. Finally, a full history of distributed alerts is maintained (**Figure 15**).

## MicroCRM

If you're running an agency and managing many servers and sites, organizing them so you can find what you're looking for quickly is important. Two aspects are built in to the MicroCRM features in WPCloudDeploy:

1. Organization features that include note taking abilities, custom links, server descriptions, labels, categories, and color coding
2. Communication features that include things like alerts, as

described previously, but also proactive notifications to interested parties associated with a server or site. For example, it is very easy to select a group of servers and sites and send off a quick message to

let interested parties know that the servers are going to be down for maintenance during xyz period. One of the most important features you'll need when you're managing a high volume of sites and servers is

Title	Server	SSH CMD
Command executed against: admin-article-server-01 — Private	admin-article-server-01 id: 81	echo "done" && ( cd ~ && sudo -E rm -rf 02-install_wordpress_site.sh &&am ...more...
Command executed against: admin-article-server-01 — Private	admin-article-server-01 id: 81	cd ~ && sudo -E rm -rf 24-server_status.sh && sudo -E wget --no-check-certificate -O ...more...
Command executed against: admin-article-server-01 — Private	admin-article-server-01 id: 81	cd ~ && sudo -E rm -rf 28-restart_callback.sh && sudo -E wget --no-check-certificate ...more...
Command executed against: admin-article-server-01 — Private	admin-article-server-01 id: 81	cd ~ && sudo -E rm -rf 24-server_status.sh && sudo -E wget --no-check-certificate -O ...more...
Command executed against: admin-article-server-01 — Private	admin-article-server-01 id: 81	sudo service php8.2-fpm status

**Figure 13:** The SSH Logs screen in WPCloudDeploy provides a full trail of what's sent and received from your server.

Title	Owner/Parent	Type	Message	Reference
admin-article-server-01 — Private	admin-article-server-01 id: 81	notice	Configuration backup has completed successfully.	backup-config
admin-article-server-01 — Private	admin-article-server-01 id: 81	notice	Configuration backup has started.	backup-config
admin-article-server-01 — Private	admin-article-server-01 id: 81	warning	The server started up.	power

**Figure 14:** The events captured by WPCloudDeploy can be used to send notifications to interested parties.

Title	Owner/Parent	Type	Message	Reference	Sent
User notification for Low Disk Space — Private	281312	alert	Email notification sent successfully for notification_id: 372042	disk-space	Yes
User notification for Low Disk Space — Private	281312	alert	Email notification sent successfully for notification_id: 372096	disk-space	Yes
User notification for Low Disk Space — Private	281312	alert	Email notification sent successfully for notification_id: 381741	disk-space	Yes
User notification for Low Disk Space — Private	281312	alert	Email notification sent successfully for notification_id: 381738	disk-space	Yes
User notification for Pending Tasks Failed Timeout (Stuck) — Private	81365	alert	Email notification sent successfully for notification_id: 338990	stuck	Yes

**Figure 15:** The history of notifications sent by email, with the use of webhooks, and to other destinations is captured for audit purposes.



the ability to find what you're looking for quickly. WPcloudDeploy includes extensive quick-filter options and robust general search capabilities on the list of servers and sites (Figure 16).

## What Else Can It Do?

WPcloudDeploy is designed for flexibility – a lot of times it errs in that direction while sacrificing the user experience. For example, you can set up servers so that each one has its own key pair, or you can conveniently use a global key pair for all servers, or you can mix and match, which adds a lot more data entry screens in both the settings area and on each server screen. The flexibility flows through many options such as backups. You can set up global backup credentials that will be used by all servers or you can set them up for each individual server.

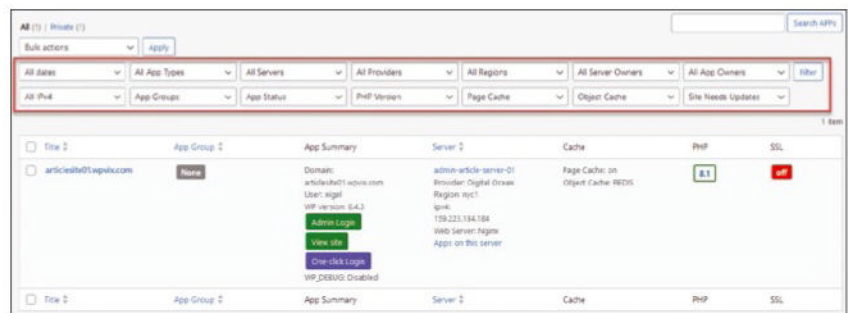
Because WPcloudDeploy is self-hosted you can deploy multiple instances, if necessary. For example, you might decide to give a dedicated panel to one of your larger customers or deploy one each in the different regions of the world where you operate.

Once you get WPcloudDeploy up and running, you should definitely dig into the Settings screen – you'll likely find a number of personalization and feature options that are easy to miss otherwise.

## What's Not Included

WPcloudDeploy has both free and premium versions. Everything discussed so far is included in the free version on GitHub. In fact, over the more than four-year life of the product, more than 80% of new features have been included in the free version.

Not included are direct connections to other cloud providers (e.g., AWS, UpCloud, Vultr, etc.), WordPress multisite networks, virtual providers (the ability to connect to multiple



**Figure 16:** WPcloudDeploy includes a set of useful predefined compound filters, as well as full-text search capabilities for certain designated fields.

accounts at the same cloud server provider), server sync, Client Power Tools (a collection of non-critical features), custom servers, and WooCommerce integration.

Two features are sort of in-between free and premium, because they are included in the free source code on GitHub but are not activated: multitenancy for WordPress and Git integration. You can activate these two features either by having a developer activate the classes in the code (which is relatively simple to do and gives you the features for free) or by purchasing them with a support contract from the WPcloudDeploy organization.

## Wrap-Up

Running WordPress sites on your own cloud servers brings a lot of benefits, including cost savings, flexibility, better privacy and security, and more. If you're running a single WordPress site, any of the WordPress pre-built server images at the major cloud providers will likely do the job for you – no special control panel required. However, when you need to run multiple sites on a server or manage dozens or hundreds of WordPress servers and sites, you need a WordPress-specific control panel that handles or masks all the WordPress idiosyncrasies.

Although you can find a number of SaaS services with WordPress-focused control panels, you will find only one option in the self-hosted

open source category that offers the full set of features: WPcloudDeploy. If you're not that picky about having a "modern" UI with all the associated glitz and glamour, you'll probably be pleasantly surprised at what you can do with it. ■

### Info

- [1] "2024's CMS Market Share Report – Latest Trends and Usage Stats," WPBeginner: [\[https://www.wpbeginner.com/research/cms-market-share-report-latest-trends-and-usage-stats/\]](https://www.wpbeginner.com/research/cms-market-share-report-latest-trends-and-usage-stats/)
- [2] WPcloudDeploy on GitHub: [\[https://github.com/WPcloudDeploy/wp-cloud-deploy\]](https://github.com/WPcloudDeploy/wp-cloud-deploy)
- [3] DigitalOcean starter image: [\[https://marketplace.digitalocean.com/apps/wpclouddeploy\]](https://marketplace.digitalocean.com/apps/wpclouddeploy)
- [4] Creating your own WPcloudDeploy starter server: [\[https://wpclouddeploy.com/documentation/wpcloud-deploy-admin/bootstrapping-a-wordpress-server-with-our-scripts/\]](https://wpclouddeploy.com/documentation/wpcloud-deploy-admin/bootstrapping-a-wordpress-server-with-our-scripts/)
- [5] DigitalOcean starter droplet docs: [\[https://wpclouddeploy.com/documentation/other-misc/digitalocean-template-image/\]](https://wpclouddeploy.com/documentation/other-misc/digitalocean-template-image/)
- [6] Latest WPcloudDeploy ZIP file: [\[https://github.com/WPcloudDeploy/wp-cloud-deploy/archive/refs/heads/main.zip\]](https://github.com/WPcloudDeploy/wp-cloud-deploy/archive/refs/heads/main.zip)
- [7] "A Watchdog for Every Modern \*ix Server" by Ankur Kumar, ADMIN, issue 77, 2023, pg. 84, [\[https://www.admin-magazine.com/Archive/2023/77/A-watchdog-for-every-modern-ix-server\]](https://www.admin-magazine.com/Archive/2023/77/A-watchdog-for-every-modern-ix-server)

### Author

**Nigel Bahadur** is product manager for the WPcloudDeploy project and has been building enterprise-level software since 1990.

# REINVENTING

# HPC



## REGISTER NOW FOR ISC 2024

Are you an HPC user, vendor, or provider?

We need your input as we embark on the mission of REINVENTING HPC.

Consider this your occasion to be a part of history in the making.

**SEE YOU IN HAMBURG!**

## DevSecOps with DefectDojo

# The Early Bird

The DefectDojo vulnerability management tool helps development teams and admins identify, track, and fix vulnerabilities early in the software development process. By Guido Söldner

**DevOps has been an integral part** of software development in most organizations for years. The term encompasses various practices and tools and a kind of cultural philosophy that are intended to help automate and interlink processes between the development department and IT teams. From DevOps mechanisms, a further development has emerged in recent years: DevSecOps, DevOps plus security. In more detail, it means that security needs to play a role in every phase of the software development process: from the initial design through integration, testing, and deployment to delivery.

The principle of moving tasks – security in this case – forward as far you can in a process chain is also known as the shift-left approach. In terms of containers, shift left means taking security aspects into account as early as the container construction stage. This approach makes sense; after all, fixing incidents in production environments often involves massive amounts of money, and discovering errors at the outset of the development process is typically far less costly. Many tools have become established on the market in the

shift-left and DevSecOps environment in recent years. DefectDojo [1] is one of these tools, and it is free.

## DefectDojo

DefectDojo was originally developed by Rackspace but is now open source. The community is working hard on the further development of the software, with more than 350 contributors and more than 2,500 GitHub Stars. New features are released quite frequently; according to the GitHub page, an update is made approximately every two weeks. The tool integrates with a wide range of existing security tools, including security scanners, issue trackers, and reporting tools and displays their information in a centralized and easy-to-understand way. A special feature is its ability to automate the process of running security scans, which makes it possible to work toward eliminating vulnerabilities in real time and to share the current status within the team. Another advantage is the tool's flexibility. DefectDojo can be customized to suit your organization's needs, with your own workflows and vulnerability

classifications, and be integrated into your own security toolchain.

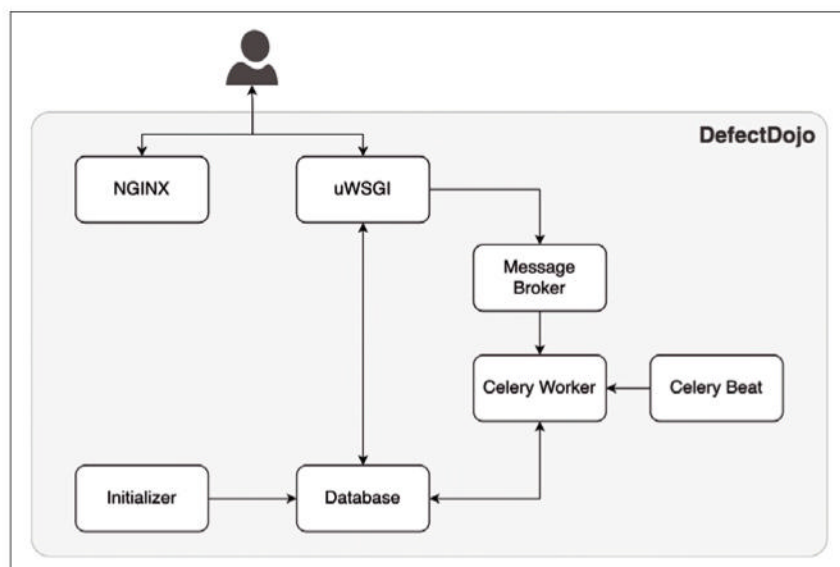
Under the hood, DefectDojo comprises a number of components (**Figure 1**):

- All static web content is provided by NGINX, including JavaScript, images, and other CSS files.
- The application server is uWSGI, which is based on the Django Python framework and is responsible for all dynamic content.
- The RabbitMQ message broker is responsible for asynchronous communication.
- Celery workers run tasks such as Jira synchronization or deduplication in the background.
- The Celery beat program is used to notify users.
- MySQL or PostgreSQL are supported as databases; PostgreSQL is recommended.
- Initializer scripts are called during the installation of updates and terminate automatically.

## Docker Installation

DefectDojo reaches the user in a containerized form; therefore, a local test install is a quick and easy process relying on Docker and Docker Compose. This environment is also a prerequisite for the installation shown in **Listing 1**. First, run the command to clone the public DefectDojo GitHub repository





**Figure 1:** DefectDojo comprises a series of interlinked open source components.

(line 1), change to the program directory (line 2), and start the local build of the container images (line 4). As soon as the build is done, you can call up the application (line 6). You need to specify a profile to save the tool's configuration. As soon as the software has launched, you will find the password in the logs (line 9); it is generated randomly for each installation. Next, open the application by calling `http://localhost:8080` and go to the dashboard. Enter `admin` as the username and the password you took from the logs.

## How DefectDojo Works

Although DefectDojo is very intuitive, it is still a good idea to familiarize

yourself with the product's data classes. The project has a clearly defined hierarchy. The Product Type occurs at the top level and is usually a company, department, or team (e.g., the Identity and Access Management team). The next level contains the matching Products (e.g., WordPress), and the third level defines a moment (Engagement) for product testing, which is usually a point in time, a version (e.g., beta), a regular security check, or the like. Each engagement has a

specific name, a time line, a leader, a test strategy, and a status.

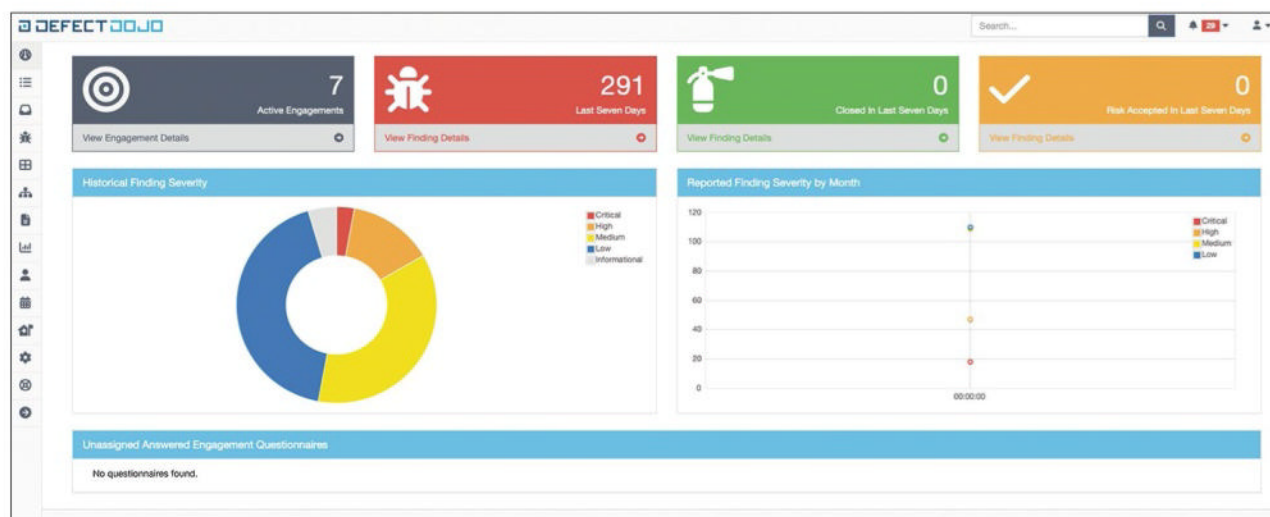
Tests summarize the activities for identifying security vulnerabilities, which are linked to a starting point, endpoint, and test type. A Finding is a vulnerability that has been found. Each finding is categorized by severity: Critical, High, Medium, Low, and Info. An example of a finding could be *OpenSSL 'ChangeCipherSpec' MiTM Potential Vulnerability*. Finally, the Endpoint designates the tested system with its IP address and fully qualified domain name.

If you use DefectDojo in a larger organization, you can easily end up with many products, engagements, tests, and other objects. To group objects, use the tags you can find in the tool's graphical user interface (Figure 2). As soon as a finding appears, its status is displayed. Each finding can be addressed individually, and you can change its status. Unfortunately, it is not uncommon for findings to appear more than once. To mitigate the effects, DefectDojo has a deduplication

### Listing 1: DefectDojo Installation

```

01 git clone https://github.com/DefectDojo/django-DefectDojo
02 cd django-DefectDojo
03 # building
04 ./dc-build.sh
05 # running (for other profiles besides postgres-redis look at https://github.com/
    DefectDojo/django-DefectDojo/blob/dev/readme-docs/DOCKER.md)
06 ./dc-up.sh postgres-redis
07 # obtain admin credentials. the initializer can take up to 3 minutes to run
08 # use docker-compose logs -f initializer to track progress
09 docker-compose logs initializer ? grep "Admin password:"
  
```



**Figure 2:** The dashboard provides an overview of the test runs (engagement) and vulnerabilities (findings).

process that can adjust the status of findings. In production environments, compliance with service-level agreements (SLAs) is typically important, as well (Figure 3). The tool also offers support; you can configure how many days software teams can take to fix findings.

DefectDojo can display the current data in the form of a report. Predefined reports exist for all data classes. If required, you can also create your own with a dedicated report builder. You can include a cover page, a directory, WYSIWYG content, findings, vulnerable endpoints, and page breaks. When it comes to visualization, metrics are particularly interesting, with product type metrics and counts or other series of figures.

## Integration with Atlassian Jira

DefectDojo thrives on integration. The Atlassian Jira [2] tool is used in many companies and can be integrated easily. Connections are possible in both directions: You can import Jira elements and return the changes to Jira. To integrate, you first need to define a webhook in Jira:

1. First open the `https://<Jira-URL>/plugins/servlet/webhooks` page in your browser.
2. Click *Create a webhook*.
3. Enter the value in the URL field: `https://<DefectDojo Domain>/`

`jira/webhook/<Webhook Secret>`. You will find the value under *Configuration | System Settings* in DefectDojo.

4. Below Comments, enable *Created* and choose the *Updated* setting for Issues.

Next, go to the *System Settings* menu in DefectDojo and click *Enable Jira integration* and *Submit*. Finally, select *Enable JIRA webhook* and press *Submit* again. You just need the granular configuration now:

1. Click *JIRA* in the menu on the left.
2. Select *Add Configuration* from the selection list.
3. Enter a username and a password. If you use JIRA Cloud, you will need the email address and an API token, as well.
4. Open `https://<Jira-URL>/rest/api/latest/issue/<any-valid issue key>/transitions?expand=transitions.fields-`
5. Enter *open status key* as the Todo id.
6. Enter *closed status key* as the Done id.
7. For admin access to Jira, open `https://<Jira-URL>/secure/admin/ViewCustomFields.jspa`; click next to *Epic Name* and then on *View*. You will then see the numerical value for the *epic name id* in the URL.

## Working with the API

If your system is not integrated, you can work with the API. Typical use cases include automated uploading

of reports from continuous integration and continuous delivery (CI/CD) pipelines (see the “Uploading from GitLab CI” box). To access the documentation, click on the user avatar in the top right-hand corner of the GUI.

Swagger definitions are also available. Before you can work with the API, it first needs to authenticate itself. The API uses an authentication header with the format *Authorization: Token <api.key>*. For example, a header token can look like:

```
Authorization: Token 80749f64ae120c27e2
504088d8c2ce29a0fa7f85c
```

The example code in Listing 2 uses Python to demonstrate how to use the API to retrieve user information. To begin, import the Python Requests API and define the URL. In the header variable, code the authorization token to call the API, and display all the elements in a for loop. The results will then look something like Listing 3. It is not always useful to display all the available elements in the output. Luckily, you can use filters (Listing 4).

## Conclusions

DefectDojo is a useful tool for centralized vulnerability management. The rich feature set and large number of plugins are major

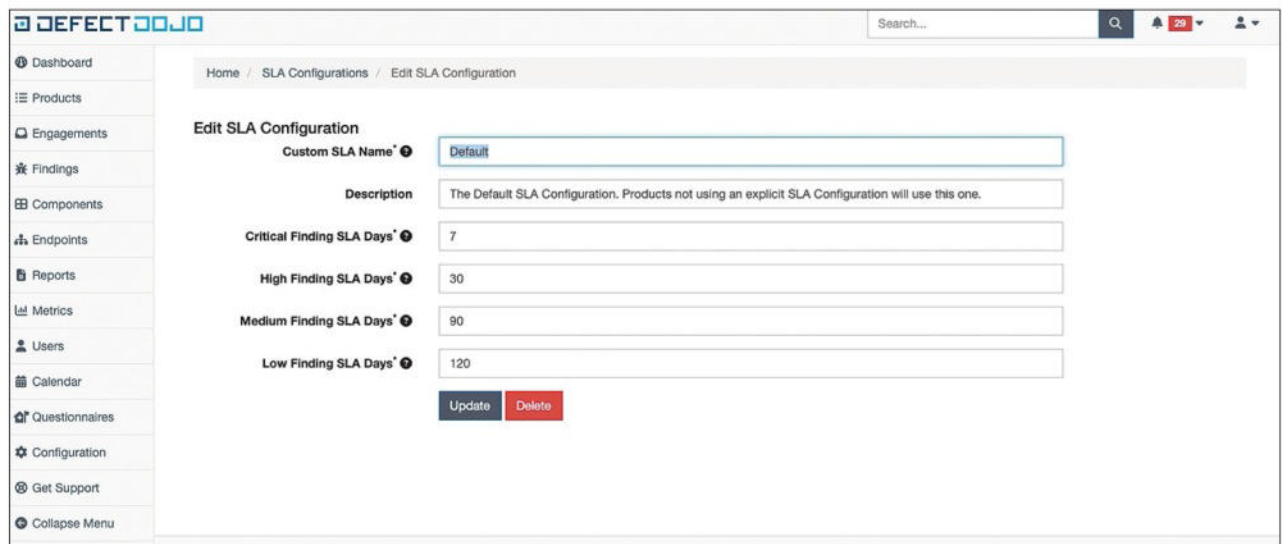


Figure 3: SLA specifications can be integrated into DefectDojo to find and fix vulnerabilities.

## Uploading from GitLab CI

In many organizations, security tools are integrated into CI/CD pipelines. For this reason, it makes sense to publish findings directly in DefectDojo, as shown with Hadolint [3] in this example. As a rule, you will always define the configuration settings in a CI script at the outset:

```
DEFECTDOJO_DIR: "."
DEFECTDOJO_HADOLINT_REPORTS: "hadolint-json-*.json reports/docker-hadolint-*.native.json"
DEFECTDOJO_BASE_IMAGE: "registry.hub.docker.com/library/node:alpine3.11"
DEFECTDOJO_NOTIFICATION_SEVERITIES: "Critical,High"
DEFECTDOJO_TIMEZONE: "Europe/Paris"
# default production ref name (pattern)
PROD_REF: '/^(master|main)$/'
DEFECTDOJO_NOPROD_ENABLED: "false"
```

Later in the script, the results from the Hadolint scan are collected:

```
# Hadolint
# template: docker
hadolint_nb_reports=0
for file in ${DEFECTDOJO_HADOLINT_REPORTS}
do
    if [[ $(expr "$file" : '.*\.*') == 0 ]] && [[ -f "$file" ]]; then
        log_info "hadolint report found: $file"
        hadolint_nb_reports=$((hadolint_nb_reports + 1))
        nb_reports=$((nb_reports + 1))
    fi
done
```

You then send these to the DefectDojo API and write the engagements:

```
_engname="Engagement ${_today_time} ${CI_COMMIT_REF_NAME} ${CI_COMMIT_SHORT_SHA}_end=${_today}
branch_tag=${CI_COMMIT_TAG}
branch_tag_info="[${CI_COMMIT_TAG}](${CI_PROJECT_URL}/-/tags/${CI_COMMIT_TAG})"
# if there is no tag, then use branch
if [[ -z "${CI_COMMIT_TAG}" ]]; then
    branch_tag=${CI_COMMIT_REF_NAME}
    branch_tag_info="[${CI_COMMIT_REF_NAME}](${CI_PROJECT_URL}/-/tree/${CI_COMMIT_REF_NAME})"
fi
dashboard_template_version=$(get_tpl_version_in_use "to-be-continuous/defectdojo")
commit_info="[commit ${CI_COMMIT_SHORT_SHA}](${CI_PROJECT_URL}/-/commit/${CI_COMMIT_SHA})\n${branch_tag_info}\ncreated with dashboard-template ${dashboard_template_version}"
echo "{\"engagement_type\": \"CI/CD\", \"product\": \"${dd_product_pk}\", \"name\": \"${_engname}\", \"source_code_management_url\": \"${CI_PROJECT_URL}\", \"commit_hash\": \"${CI_COMMIT_SHA}\", \"branch_tag\": \"${branch_tag}\", \"status\": \"In Progress\", \"target_start\": \"${_today}\", \"target_end\": \"${_end}\", \"description\": \"${commit_info}\"}" >
api_input.json
# post request to create engagement
curl -LX POST -d @api_input.json "${DEFECTDOJO_SERVER_URL}/api/v2/engagements/" --header
"Content-Type: application/json" --header "Authorization: Token ${DEFECTDOJO_API_KEY}" --verbose
1> api_output.txt
engagement_id=$(jq ".id" api_output.txt)
echo "engagement_id: $engagement_id"
if [ "${hadolint_nb_reports}" -gt 0 ]; then
    docker_tpl_version=$(get_tpl_version_in_use "to-be-continuous/docker")
    log_info "Docker template version: $docker_tpl_version"
    for file in ${DEFECTDOJO_HADOLINT_REPORTS}
    do
        if [[ $(expr "$file" : '.*\.*') == 0 ]] && [[ -f "$file" ]]; then
            import_scan "$file" "Hadolint Dockerfile check" "$engagement_id"
            "to-be-continuous/docker ${docker_tpl_version}"
        fi
    done
fi
# Close the engagement
curl -L -X POST "${DEFECTDOJO_SERVER_URL}/api/v2/engagements/$engagement_id/close/" --header
"Authorization: Token ${DEFECTDOJO_API_KEY}" -d ''
curl -L
"${DEFECTDOJO_SERVER_URL}/api/v2/findings/?test_engagement__product=${dd_product_pk}&severity=${DEFECTDOJO_NOTIFICATION_SEVERITIES}&limit=100&false_p=false&duplicate=false&active=true" --header "Content-Type: application/json" --header "Authorization: Token ${DEFECTDOJO_API_KEY}" --verbose 1> api_final_findings.json
```

advantages. If required, you can use the API or extend the product. The orchestration options with CI pipelines make it possible to publish vulnerabilities directly (e.g., when building containers). DefectDojo's scalability is also noteworthy and offers good support for large teams. ■

## Info

- [1] DefectDojo: [\[https://www.defectdojo.org\]](https://www.defectdojo.org)
- [2] Atlassian Jira: [\[https://www.atlassian.com/software/jira\]](https://www.atlassian.com/software/jira)
- [3] Hadolint: [\[https://hub.docker.com/r/hadolint/hadolint\]](https://hub.docker.com/r/hadolint/hadolint)

## Listing 2: Retrieving User Information

```
01 import requests
02 url = 'http://127.0.0.1:8080/api/v2/users'
03 headers = {'content-type': 'application/json', 'Authorization': 'Token
04 80749f64ae120c27e504088d8c2ce29a0fa7f85c'}
05 r = requests.get(url, headers=headers, verify=True)
06 # set verify to False if ssl cert is self-signed
07 for key, value in r.__dict__.items():
08     print(f'{key}: {value}')
09     print('-----')
```

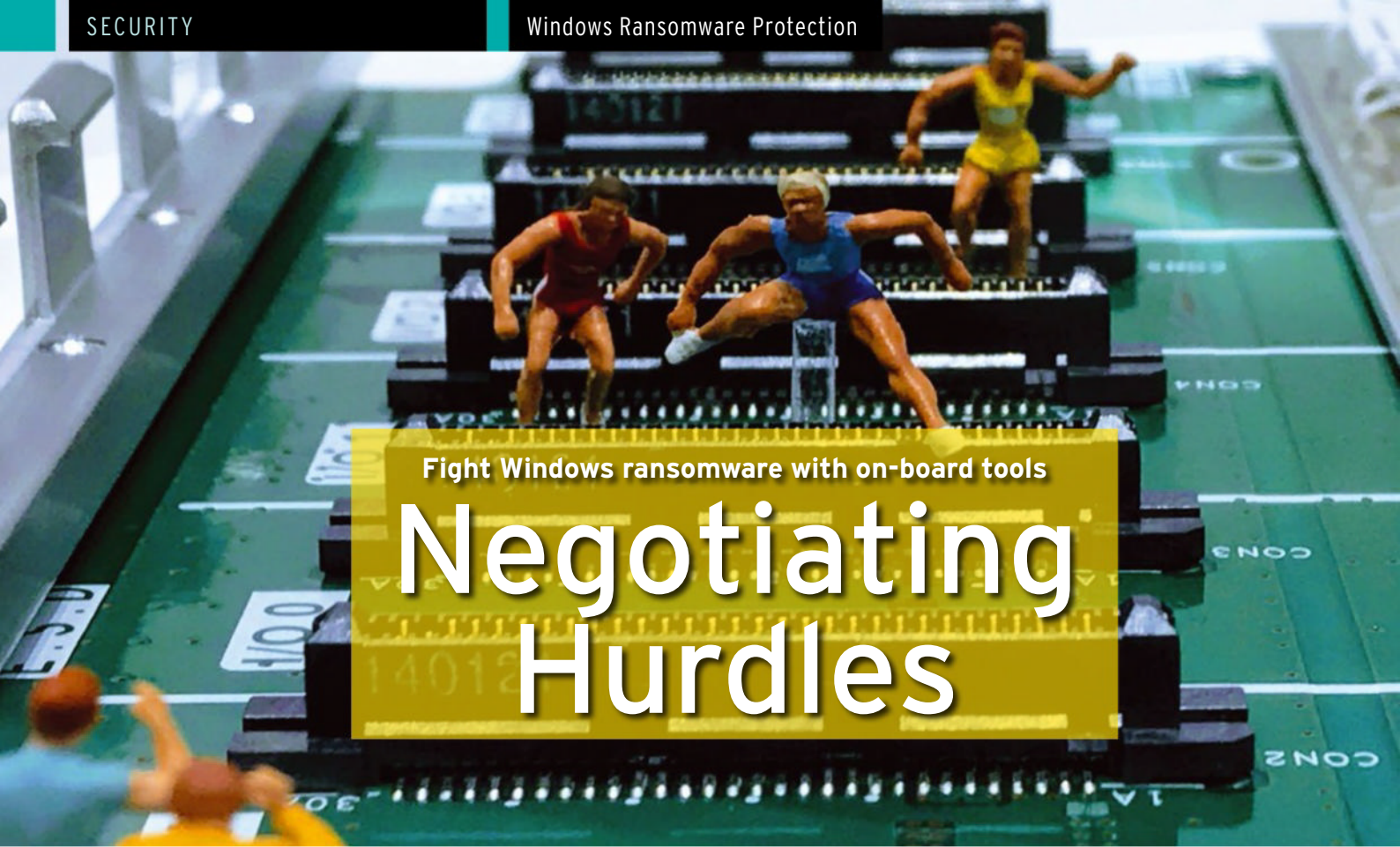
## Listing 3: Return User Info as JSON

```
01 [
02     {
03         "first_name": "Tyagi",
04         "id": 22,
05         "last_login": "2019-06-18T08:05:51.925743",
06         "last_name": "Paz",
07         "username": "dev7958"
08     },
09     {
10         "first_name": "saurabh",
11         "id": 31,
12         "last_login": "2019-06-06T11:44:32.533035",
13         "username": "saurabh.paz"
14     }
15 ]
```

## Listing 4: Filter with for Loop

```
01 import requests
02 url = 'http://127.0.0.1:8080/api/v2/users/?username_contains=jay'
03 headers = {'content-type': 'application/json', 'Authorization': 'Token
04 80749f64ae120c27e504088d8c2ce29a0fa7f85c'}
05 r = requests.get(url, headers=headers, verify=True)
06 # set verify to False if ssl cert is self-signed
07 for key, value in r.__dict__.items():
08     print(f'{key}: {value}')
09     print('-----')
```





Fight Windows ransomware with on-board tools

# Negotiating Hurdles

Ransomware defense involves two strategies: identifying attacks and slowing the attackers to mitigate their effects. By Mark Heitbrink

**The horror scenario:** Your organization's data has been encrypted – in the worst case, after the data has been stolen and is at risk of ending up on the darknet. The measures used to mitigate the effect of ransomware can be broken down into two aspects. The first involves preventing attacks, and the second is all about slowing down the attack if it is successful. Both tasks require changes to workflows and processes involving administrative intervention that is not always convenient.

## Entry

Ransomware has a limited number of vectors for entering the company network. Email and malicious attachments come first, but external access to the mailbox is also conceivable, with the manipulation of existing attachments. Many companies also have holes in the firewall that provide a direct route to the internal network. Remote Desktop (RDP) and other protocols that allow remote access are worthy of note, as well as manipulated software that users download and install. Last but not least, one visit to a

manipulated website is all it takes to be infected by ransomware or some other malware (drive-by attacks).

## Email

Email is the most common way for ransomware to enter a company. A simple file attachment is all it takes. Sending billions of email messages costs nothing but electricity. Valid target addresses can be bought, found, and generated. Anyone who has worked with the same email address for a period of time will be familiar with the problem of spam and be aware that their own address has been public knowledge for a long time. What was technically brilliant about the Locky attack [1], for example, was that the malware and the associated executable file were not directly included in the attachment. Instead, the recipients received an Excel file with a macro that acted as a downloader. The executable was only downloaded from the Internet and executed when the macro was executed. Virus scanners, especially those on mail servers, did not sound the alarm because the attachment itself did not appear to be critical.

You are playing a constant game of cat and mouse: Attackers create malware that will sooner or later be detected by antivirus programs. Because of the abundance of malicious code and the increasingly clever tricks used by attackers to disguise their malware, antivirus (AV) manufacturers have had to switch to behavior-based detection. If a file originates from the Internet, is not digitally signed by a trustworthy software manufacturer, and possibly attempts to access critical system areas, the alarm bells go off. The AV tools also block suspicious connections to the Internet. In this race, the attackers usually come out on top. If you create a rule that prohibits Excel files with macros, the attackers switch to PDF files with JavaScript or HTML files with an encapsulated script. OneNote files containing Excel files with macros are also a potential vehicle. The game goes on endlessly, and at the end of the day, attachments with dangerous content will always slip through. One relatively simple method to manage this problem is to ban blanket acceptance of email messages with attachments. If no attachments come in, no spam filter or virus scanner has to evaluate and recognize them. A workflow in the form of a fixed process is required to receive legitimate

email with attachments. Third-party providers can help. Anyone wanting to send an email with an attachment needs to communicate with the recipient beforehand. The recipient can negotiate a route with the sender through which the attachment can be transferred. After all, a company does not want to lose an application. zip file if an application comes from a legitimate source. If you want to send something, you can be expected to respond to a reply email telling you to contact the receiver personally. Third-party providers in this mail flow have ready-made dialogs and portals in their portfolio that initiate the upload with password protection. The password can be trivial and is negotiated spontaneously by telephone; the upload is only permitted during a specific time window. The attachment can then be examined and analyzed before it finds its way to the recipient. Of course, automated systems must have allowlists, but normal users in your company and their counterparts can usually be expected to handle a few extra clicks. The advantage of this workflow is that a company does not have to react to special extensions, but simply blocks everything that comes in without a prior agreement.

## Insecure Passwords

Another construction site in IT security is passwords. The sad truth is that companies still use insecure passwords and don't dare switch to at least 16 characters because users can't be expected to remember that many. However, it must be clear to all companies that direct external access to any system over the Internet can no longer be protected by a moderately secure password alone. Even a 20-character password can fail if it falls into the hands of attackers. Therefore you need multifactor authentication (MFA), one-time passwords (OTPs), and similar procedures. Many companies have opted for the cloud. Exchange Server is often exposed on the network and can be accessed from anywhere in the world. Imagine the only protection for the managing director's email account

being the password *Summer23*, because the managing director has to change the password every three months and doesn't feel like doing so. The annoying thing is that *Summer23* complies with the Microsoft complexity rule for passwords: upper and lower case letters, numbers, and special characters – three of these four character sets are required – along with a minimum length of six characters. This password would have any reasonably experienced attacker laughing out loud. As soon as an attacker gains access to the mailbox, they can, for example, reply to a co-worker's email containing an Excel spreadsheet with another Excel spreadsheet with macros and malicious code. In this case, the recipient will never suspect an attack, because it is a direct reply to their own email from somebody they know. Moreover, corporate policies for internal attachments are often more lax than for external posts – a phenomenon that the notorious Emotet automated banking trojan exploited. Another option that attackers now often use is manipulated invoices. Emailing invoices is an established practice. If the dispatched PDF is redirected to the attacker by a mail rule in Outlook, the PDF can be provided with a different invoice total, and the bank details can be adjusted so that it can then be sent out again by someone who looks like the legitimate sender. Ultimately, attachments will always find a way into the company. As soon as employees open these attachments and malicious code is executed, companies rely on their virus scanners and combine them with behavior-based detection – especially in a cloud with an endpoint (E), managed (M), or extended detection and response (XDR) service. Providers can make good money by offering their protection as a subscription and bill per endpoint. The organization relinquishes responsibility and relies on a tool to detect incidents.

## Allowing Specific Software

Wouldn't it be far better to retain control and, above all, not let attacks

happen in the first place? A number of strategies are at hand. Although not a popular approach, allowlisting and associated tools have been around since time immemorial:

- XP, Service Pack 2, software restriction policies (aka SAFER)
- Windows Vista, AppLocker (Enterprise, aka SAFER2)
- Windows 10 1607, Device Guard
- Windows 10 1709, Device Guard becomes Windows Defender Application Control (WDAC)
- Patchday 30 September 2022: Every Professional version of Windows now supports AppLocker with Group Policy; it is no longer an Enterprise feature. Although you have always been able to set up AppLocker by mobile device management (MDM) in the Professional versions, the process has been quite unwieldy.

Blocking unknown software is certainly the sharpest sword in the defense arsenal, similar to completely blocking email attachments. In principle, only code known to the company can be executed. In other words, malicious code has to be very carefully crafted if it is to work. You might still have gaps in the rules, but if so, you should establish software allowlisting in the enterprise and maintain it as an ongoing process.

Understandably, administrators are reluctant to add this work to the usually huge zoo of custom applications. However, if you simplify the approach and declare only `%programfiles%` and `%systemroot%` to be secure, because these paths cannot be changed in the user context, you have already provided protection against most malware with just two rules. Of course, these actions will not protect against attackers with admin rights, but the first line of defense has to withstand most attacks. Users have no write permissions for these paths, with just a few exceptions that can also be regulated [2].

## Restricting Access to Folders and PowerShell

Malware usually needs a location with write access to generate an

executable or to save the script. Everything below %userprofile% and C:\Temp, both of which are easily accessible directories, is addressed. It is therefore important only to allow paths for execution to which a user does not have write access. In the phenomenon of “dummy folders,” the Windows Explorer API does not distinguish between “C:\Windows” and “C:\Windows ” (with a space at the end). However, code can be stored in “C:\Windows ” and Explorer will think it is talking to C:\Windows. Annoyingly, Windows allows users (and therefore also attackers) to create a new directory directly below C:, which must be prevented at the NTFS level [3].

In addition to path rules, publisher certificates can be used to authorize software. Programs that do not have a certificate can be self-signed if necessary with your own code signing certificate and the Set-AuthenticodeSignature cmdlet. If you have a code-signing certificate, you can also use it to sign macros and PowerShell scripts that are allowed to run. Only allow certain paths for execution, and only trust a few certificates – this is a clear set of rules. Of course, you can go one better: You can target specific program versions or create hashes for individual files. Initially, however, a less granular, simpler set of rules should be established to allow the technology to be used.

PowerShell must also be regulated for users, because it is ultimately an admin tool. Among other things, it also allows code to be reloaded or executed without files [4].

## Getting Started with AppLocker

One criticism of Microsoft’s own product is the poor and sometimes painstaking administration; on the upside, it

is available as an on-board tool and can be used directly. AppLocker (Figure 1) has a monitoring function that only logs events prior to enabling the feature. Monitoring needs to be combined with Windows event forwarding to collect the events of blocked applications in a central location. However, the idea that you can start AppLocker today and be ready to go after a few days is misguided. Instead, you start monitoring and improving your set of rules, and once the errors stop, the system switches from monitoring to enforcing. Depending on the company and the test clients used, this process can take two weeks or two months. The important thing is for the IT department to get started. Third-party providers usually use intelligent agents on the clients that also support port blocking in addition to software allowlisting and better central management. In AppLocker’s favor, however, is that you have no excuse not to start using it this very moment. Without question, a virus scanner or the Microsoft Defender attack surface reduction rules also need to be in place, but allowlisting is the most important part. Unlocking software in a targeted way also shuts down two further vectors: unwanted software downloads and malware on websites. Software is often not downloaded directly from the manufacturer but by dubious portals that are likely to be at the top of the search engine list. The other goodies that end up on the computer from these sites is

unknown. Another useful side effect of allowlisting is gaining control over the Microsoft Store; admins have been tearing their hair out over this repository for years. The answer is simple: Allow the Windows Store and use a publisher rule to control the apps. What is not permitted will simply not happen. It is sometimes irritating to see how companies refuse to use this technology because they are afraid of extra work and administrative overhead.

Before I forget, having a potentially unwanted application (PUA) or potentially unwanted program (PUP) also means extensions in web browsers, which might have direct access to the form data of websites and therefore to password fields, bank details, and so on. Not every extension does only what you have downloaded it for. The three major browsers – Chrome, Edge, and Firefox – offer group policies to control extensions. A blocklist (\*, everything) is implemented and only what is known is allowed. Conveniently, the extensions have unique identifiers which can be used to install extensions or allow them to be installed automatically.

## Segmentation at the Admin Level

The company is not a perfect fortress. Despite a wide range of measures, malware can still break in. Depending on the type of attack, you can suffer a short, hard punch,

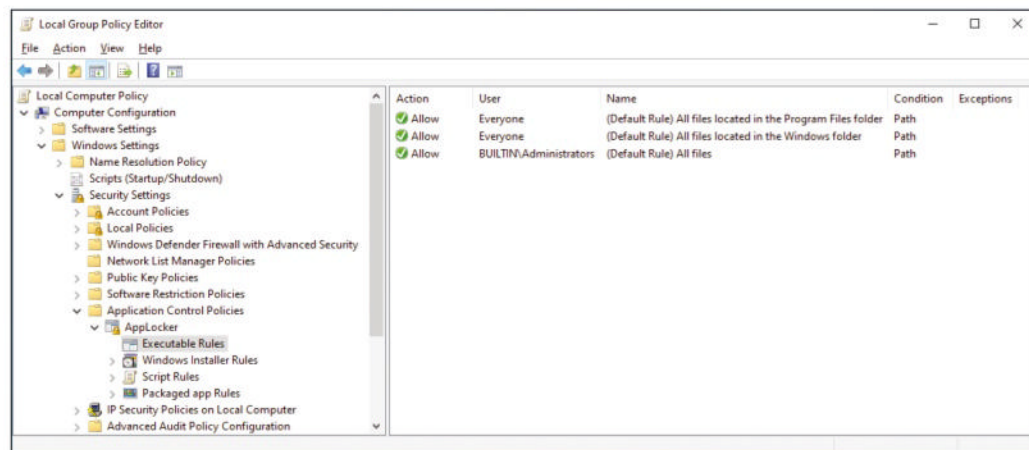


Figure 1: AppLocker is Microsoft’s (admittedly inconvenient) approach to allowing specific software.



in which the attacker gets into the company, encrypts as much as they can in the shortest possible time, and demands a ransom, or a slow but steady settling in, wherein the attackers work their way up to the crown jewels of the company, including hijacking admin accounts. The slow approach is a means to an end that provides access to interesting targets.

Administrators need to change the way they work and recognize that, instead of wielding a master key for company IT, they need a key ring: It's all about delegation and tiering. At present, many admins still work with a domain admin account when carrying out their day-to-day business of accessing clients, servers and, of course, the domain controller. This account is very convenient, because domain admins are members of the local group of administrators on every Microsoft system by default user mapping. The members of the local groups on a system can be regulated by group policy. The scope of an administrative account must be kept as tight as possible.

If a client has been hacked and a highly privileged account logs on to this computer, the attacker can hijack the session or use a keylogger to capture the password. High-value admin

accounts must therefore be denied login and access to compromised systems.

The first step is usually to break accounts down into client, server, and domain admins. A client admin authorized to manage all clients is still a potential target that is not much smaller than that of a domain admin. After all, they manage all the clients. The solution is to break the accounts down further to the level of individual systems. Microsoft offers the Local Administrator Password Solution (LAPS) for this purpose. From May 2015 to March 2023, an additional piece of software was required, but LAPS became a built-in system component in April 2023 [5] (Figure 2). The tool generates an individual password for a local admin account and documents it in the respective computer object in Active Directory (AD). The scope is therefore limited to the one client, and the password is only valid on the one machine. Likewise, not every administrator needs to have domain admin rights just because they have to add a computer to AD or reset a password. The directory has always allowed delegation, and consequently rights assignments, similar to the NTFS permissions in the filesystem – but

in this case by way of organizational units in the form of levels and objects. Protecting accounts through this type of separation is essential and, incidentally, not a new idea [6]. Microsoft published two PDF files back in 2012 explaining the underlying technology and offering help with the necessary measures [7].

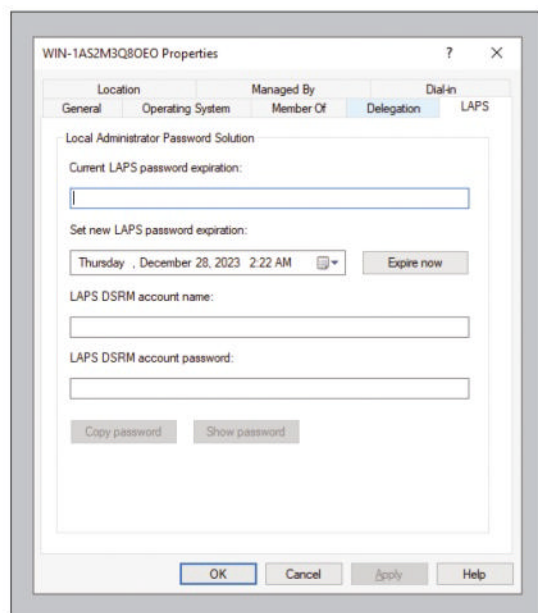
## Conclusions

The safeguards for your network must be designed such that attackers are slowed, possibly losing interest, while

you endeavor to identify attacks. Segmentation of administrative accounts goes hand in hand with network segmentation. The aim is to prevent lateral movement on a peer level and, above all, escalation into privileged areas. The Windows firewall can be a huge help in combination with group policies. If you manage to prevent a normal client from talking to other clients, then you have already gained a good deal of ground [8]. In addition to the major construction sites mentioned in this article, many other aspects also need to be considered, including patch management and employee training, along with rethinking backup strategies. ■

### Info

- [1] Locky: [\[https://en.wikipedia.org/wiki/Locky\]](https://en.wikipedia.org/wiki/Locky)
- [2] AppLocker vs software restriction policy: [\[https://serverfault.com/questions/447078/applocker-vs-software-restriction-policy\]](https://serverfault.com/questions/447078/applocker-vs-software-restriction-policy)
- [3] “ ‘Mock Folders’ as UAC bypass security disaster, leverage AppLocker and SRP” by Günter Born, Born's Tech and Windows World, March 11, 2023: [\[https://borncity.com/win/2023/03/11/windows-10-11-mock-folders-as-uac-bypass-security-disaster-leverage-applocker-and-srp/\]](https://borncity.com/win/2023/03/11/windows-10-11-mock-folders-as-uac-bypass-security-disaster-leverage-applocker-and-srp/)
- [4] Blocking PowerShell for users: [\[https://www-gruppenrichtlinien-de.translate.google/artikel/powershell-fuer-benutzer-verbieten?\\_x\\_tr\\_sl=de&\\_x\\_tr\\_tl=en&\\_x\\_tr\\_hl=en&\\_x\\_tr\\_pto=wapp\]](https://www-gruppenrichtlinien-de.translate.google/artikel/powershell-fuer-benutzer-verbieten?_x_tr_sl=de&_x_tr_tl=en&_x_tr_hl=en&_x_tr_pto=wapp) (in German)
- [5] Migrating from LAPS Legacy to LAPS Native: [\[https://www-gruppenrichtlinien-de.translate.google/artikel/migration-laps-legacy-zu-laps-nativ?\\_x\\_tr\\_sl=de&\\_x\\_tr\\_tl=en&\\_x\\_tr\\_hl=en&\\_x\\_tr\\_pto=wapp\]](https://www-gruppenrichtlinien-de.translate.google/artikel/migration-laps-legacy-zu-laps-nativ?_x_tr_sl=de&_x_tr_tl=en&_x_tr_hl=en&_x_tr_pto=wapp)
- [6] Privileged access model: [\[https://learn.microsoft.com/en-us/security/privileged-access-workstations/privileged-access-access-model\]](https://learn.microsoft.com/en-us/security/privileged-access-workstations/privileged-access-access-model)
- [7] Protection against credential theft: [\[https://www.microsoft.com/en-us/download/confirmation.aspx?id=36036\]](https://www.microsoft.com/en-us/download/confirmation.aspx?id=36036)
- [8] Defender Firewall with advanced security: [\[https://www-gruppenrichtlinien-de.translate.google/artikel/microsoft-defender-firewall-mit-erweiterter-sicherheit?\\_x\\_tr\\_sl=de&\\_x\\_tr\\_tl=en&\\_x\\_tr\\_hl=en&\\_x\\_tr\\_pto=wapp\]](https://www-gruppenrichtlinien-de.translate.google/artikel/microsoft-defender-firewall-mit-erweiterter-sicherheit?_x_tr_sl=de&_x_tr_tl=en&_x_tr_hl=en&_x_tr_pto=wapp)



**Figure 2:** Thanks to LAPS, admin passwords can no longer be stolen so easily from managed clients.



Want to get ADMIN in your inbox?

**Subscribe free to  
ADMIN Update**

and get news and technical articles  
you won't see in the magazine.

[bit.ly/HPC-ADMIN-Update](https://bit.ly/HPC-ADMIN-Update)

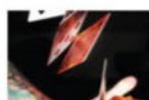
## ADMINUPDATE

January 10, 2024

Issue 361



### This Week's Feature



#### Response Automation with Shuffle

Security orchestration, automation, and response (SOAR) is increasingly important to counter ever-growing IT security threats. Shuffle lets you define automated workflows that boost infrastructure security.

### News and Resources

- [How Vector Databases Work](#)
- [Microsoft Introduces Copilot Key to PC Keyboards](#)
- [CISA Warns of Vulnerabilities Affecting Google Chromium WebRTC and Excel](#)

### In Case You Missed It

#### A Modern Logging Solution

Fluentd and its lighter counterpart Fluent Bit can help you unify data collection and consumption to make sense of logging data.



# Get to know **ADMIN**

*ADMIN Network & Security* magazine  
is your source for technical solutions  
to real-world problems.

*ADMIN* is packed with detailed discussions aimed  
at the professional reader on contemporary  
topics including security, cloud computing,  
DevOps, HPC, containers, networking, and more.

**Subscribe to *ADMIN***  
and get 6 issues  
every year



@adminmagazine



@adminmag



ADMIN magazine



@adminmagazine





## Secure Kubernetes with Kubescape

# Inspection

Kubescape checks Kubernetes container setups for security and compliance issues, making life easier for administrators.

By Martin Loschwitz

**Containerized environments** are complex and comprise several layers, especially if Kubernetes is involved as a fleet orchestrator. Container security is a particular challenge because today's cloud stack combines so many components from so many different sources, in a more or less meaningful way, that it is not easy to keep track of and identify security updates for the various sources, finding the ones that you need for your own environment, and installing them in good time. As if that weren't enough trouble, more or less the same thing applies to compliance. Most glaring security issues are not caused by bugs, but by trivial misconfigurations that nobody notices in the review. If all the internal control processes fail,

your own container landscape is left as open as the proverbial barn door in a worst case scenario.

To ensure that containerland does not turn into a horror movie, companies need to bear a few things in mind when they look to operate a large number of containers. After all, Kubernetes and others of the same ilk do not maintain themselves, and container-based approaches are no less complex than their traditional predecessors; you have to deal with even more loose ends than in conventional setups. The runtime environment for containers, Kubernetes itself, a number of on-top solutions such as the Istio service mesh, various package managers such as Helm, and the various sources from which container

images can be obtained today are just a few examples.

This is where Kubescape [1] enters the scene. Its developers make some bold promises, claiming that it is the first tool that can completely automate the process of checking the entire container stack of an environment for security and compliance problems according to accepted rules (e.g., from the US National Institute of Standards and Technology (NIST), the not-for-profit MITRE organization, or the joint US National Security Agency (NSA)-Cybersecurity and Infrastructure Security Agency (CISA)). Kubescape not only looks at the basic services that belong to Kubernetes itself but also checks YAML files with resource definitions, code in directories (e.g., GitHub or GitLab), the artifacts that arise from continuous integration and continuous delivery (CI/CD) tools such as Jenkins or Argo, and the deployment itself and associated components.

## Complex Kubernetes Setups

According to the Kubescape developers, all you have to do is install, launch, and be happy. Kubescape

Lead Image © iimbi007, 123RF.com

really does an amazing job, and once you have familiarized yourself with the program, you will be thrilled with the versatile feature set it offers. What's more, Kubescape is free software, an official Cloud Native Computing Foundation (CNCF) sandbox project freely available online, which is reason enough for Kubernetes administrators to take a closer look at the tool and try it out.

Before you do, however, you need to look at a bit of theory – after all, Kubescape is also complex under the hood. To make sure you don't end up with yet another tool whose functionality you can only guess at in a vague way, you need to take a closer look at the capabilities of Kubescape, which, in turn, means taking a closer look at the structure of a standard Kubernetes setup. In this way you can identify the components that play a role in terms of security and compliance and see where Kubescape enters the scene when you want to monitor these components.

The components of Kubernetes (K8s) are defined and obvious, as are the helpers that K8s needs to operate containers. They include the Kubernetes API, the scheduler, the K8s controller cluster, and the agents (aka kubelets) on the target

systems. Moreover, K8s is practically useless without a runtime environment for containers on a compute node, which means you need either the Docker Community Edition or Podman, which also offers the compatible CRI-O runtime environment. Together, these components are all it takes to manage containers across the boundaries of individual compute systems.

In the vast majority of today's environments, though, these basic components are not enough because they only offer very basic functionality for complex software-defined networking (SDN) and software-defined storage. Without these components, container fleets in particular cannot be operated meaningfully. After all, if you are building a scalable platform, you will also need a scalable network and scalable storage.

More components are quickly added, such as Calico container and network security, or Rook, which bundles the Ceph object storage solution into K8s and makes it manageable. Both Rook and Calico make extensive use of custom resource definitions, which, strictly speaking, must be handled as a separate factor in the security context; even a hardened Kubernetes is useless if it becomes vulnerable because of

the extensions you added or just installed.

The prebuilt Rook solution is by no means the only project that manipulates Kubernetes from the outside. Tools such as Istio are also likely candidates, extending the Kubernetes API with their own settings and services (Figure 1). To make matters worse, all of these tools and extensions come with their own software in their own containers, and you need to at least monitor the containers to keep your environment secure.

No mention has yet been made of tools that you add to your setup in other ways. For example, the Helm package manager promises to facilitate software installation tasks but itself consists of software that is potentially susceptible to security problems. Remember that, in a Kubernetes cluster, all these challenges are just the icing on top of those that come with your normal systems.

Containers do not run in a vacuum – they require standard Linux systems with a runtime environment, which can also be affected by security issues or incorrect settings. The trend in container environments is to degrade the host system to a container playback box, but you also have a Linux kernel and a basic set of userland



```
apiVersion: v1
kind: Service
metadata:
  name: {{ include "common.names.fullname" . }}
  namespace: {{ .Release.Namespace }}
  labels: {{- include "common.labels.standard" . | nindent 4 }}
    {{- if .Values.commonLabels }}
    {{- include "common.tplvalues.render" ( dict "value" .Values.commonLabels "context" $ ) | nindent 4 }}
    {{- end }}
    {{- if .Values.commonAnnotations }}
  annotations: {{- include "common.tplvalues.render" ( dict "value" .Values.commonAnnotations "context" $ ) | nindent 4 }}
  {{- end }}
spec:
  type: {{ .Values.service.type }}
  sessionAffinity: {{ default "None" .Values.service.sessionAffinity }}
  {{- if (and .Values.service.clusterIP (eq .Values.service.type "ClusterIP")) }}
  clusterIP: {{ .Values.service.clusterIP }}
  {{- end }}
  {{- if (and .Values.service.loadBalancerIP (eq .Values.service.type "LoadBalancer")) }}
  loadBalancerIP: {{ .Values.service.loadBalancerIP }}
  {{- end }}
  {{- if (and (eq .Values.service.type "LoadBalancer") .Values.service.loadBalancerSourceRanges) }}
  loadBalancerSourceRanges: {{- toYaml .Values.service.loadBalancerSourceRanges | nindent 4 }}
  {{- end }}
  {{- if (or (eq .Values.service.type "LoadBalancer") (eq .Values.service.type "NodePort")) }}
  externalTrafficPolicy: {{ .Values.service.externalTrafficPolicy | quote }}
  {{- end }}
```

**Figure 1:** Kubernetes comprises several layers and can be extended externally with complex service definitions, such as from Istio, as shown here. Ensuring security and compliance is a highly complex task.

software, which can cause worries in terms of security.

## Kubescape Scans Everything

Kubescape enters the scene with the promise of making your life easier in this complex situation by applying defined best practices and screening your entire environment in a completely automated way. The operating principle is simple: Download the tool. For a change, Kubescape does not have to run as a Kubernetes component to perform its task. It is fully external and therefore avoids the criticism levied at many other Kubernetes tools that it only sees things “from the inside” and cannot identify problems outside of the Kubernetes universe.

Kubescape scans everything – resources in Kubernetes, the container images you use, and the host system configurations with any software installed there. At the end of a run, it displays an overview of the security and compliance problems found and sorts its findings according to how threatening they are. Clear and urgent security problems are highlighted in red and are found at the

top of the list of results. Results where Kubescape is not entirely certain, but which – if true – would also be problematic, are highlighted in yellow in the middle of the list. Less important entries follow at the end.

Kubescape also applies a point system, awarding points for each problem it finds. If an installation exceeds a defined point limit, Kubescape sounds the alarm and prompts you to take action. In the default configuration, critical vulnerabilities will always mean that the number of points exceeds the alert threshold. If this principle reminds you of Chef’s InSpec, well spotted. You would not be mistaken to describe Kubescape as a technology twin of InSpec that is specialized for use in Kubernetes environments.

## Installing Kubescape

After all this theory, you still need to know how to put Kubescape to practical use, and it’s not as complicated as you might expect, given the range of functions I just described. For a smooth start, your working environment needs to fulfill only a few conditions. The first is almost

self-explanatory: You must have access to the entire K8s instance from the system on which you want to run Kubescape. If not already in place, you are advised to set up something like a cluster workstation that can access the entire Kubernetes cluster and the host systems.

In many places, IT and security departments tend to hide setups behind as many ridiculous firewall constructs as possible; and in some cases, this even means having firewalls between the individual systems in an environment. If this is true in some of your cases, you might need to use more than one host to perform scans. Of course, the host for Kubescape operation can be virtual. How you install Kubescape essentially depends on your personal preferences. The easiest way is to download the program directly from GitHub onto any Linux system:

```
curl -s https://raw.githubusercontent.com/2
kubescape/kubescape/master/install.sh | 2
/bin/bash
```

The developers expressly point out that Kubescape is a security solution. For compliance reasons alone, you

```
k8suser@wslmicron:~$ echo $KUBECONFIG

k8suser@wslmicron:~$ ls $HOME/.kube
ls: cannot access '/home/k8suser/.kube': No such file or directory
k8suser@wslmicron:~$ kind create cluster --name wslkind
Creating cluster "wslkind" ...
 ✓ Ensuring node image (kindest/node:v1.17.0)
 ✓ Preparing nodes
 ✓ Writing configuration
 ✓ Starting control-plane
 ✓ Installing CNI
 ✓ Installing StorageClass
Set kubectl context to "kind-wslkind"
You can now use your cluster with:

kubectl cluster-info --context kind-wslkind

Have a nice day! 🍀
k8suser@wslmicron:~$ ls $HOME/.kube
config
k8suser@wslmicron:~$ kubectl cluster-info
Kubernetes master is running at https://127.0.0.1:32772
KubeDNS is running at https://127.0.0.1:32772/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
k8suser@wslmicron:~$
```

**Figure 2:** For Kubescape to work, you need to store the kubeconfig file, which is also the basis for kubectl. Installation tools such as kind create the file automatically.



will want to download and investigate the shell script referenced by this command before running it. As an alternative, prebuilt packages are available for a number of Linux distributions. If you want to run Kubescape on a recent Ubuntu system, for example, you will find complete packages in a Launchpad PPA directory:

```
sudo add-apt-repository 2
    ppa:kubescape/kubescape
sudo apt update
sudo apt install kubescape
```

Make sure this directory is enabled on the system, and install the package required for Kubescape. This approach ensures that updates in Launchpad automatically find their way into your installation, which is not the case with the shell script variant described first.

The approach for RHEL is somewhere in between; the developers do not offer a usable repository, but at least you can use the prebuilt RPM packages. The packages are available

online [2] and can be installed with `dnf`. Again, you are responsible for handling the updates yourself.

## Granting Access to Kubernetes Clusters

Once Kubescape is ready to launch on the local system, simply typing `kubescape` should display a help text on the screen. Without further preparations, you would simply see an error message if you call the

```
kubescape scan --verbose
```

command. For Kubescape to examine a Kubernetes cluster, it needs to know where the nodes are and, of course, access them. The Kubescape developers' implementation is exemplary; they use the same cluster configuration that is also used for `kubectl` – the main management tool for K8s in general. However, if you set up your own host for Kubescape operation, as recommended, you will probably not have this configuration.

The quick fix for the problem if you already have a system with `kubectl` running is simply to copy the entire `~/.kube/` folder from that system to the Kubescape host, including `~/.kube/config`, which contains the access data for Kubernetes. The `kubeconfig` files are produced by K8s at least once during the initial setup of the cluster (Figure 2); for existing K8s systems, you will – at the least – find it on the host on which the Kubernetes setup was carried out. Tools such as OpenShift often also display the `kubectl` configuration on the screen during their setup, allowing it to be backed up independently. The chances are basically good that the file is available on a system if the call to

```
kubectl cluster-info
```

works. Incidentally, `kubectl` should also exist on the host on which you later call `kubescape`. The K8s developers explain how to do this in their instructions [3]. In most cases, however, you just need to install the

Controls: 65 (Failed: 21, Passed: 34, Action Required: 10)  
Failed Resources by Severity: Critical – 0, High – 0, Medium – 95, Low – 58

SEVERITY	CONTROL NAME	FAILED RESOURCES	ALL RESOURCES	% COMPLIANCE-SCORE
Critical	Disable anonymous access to Kubelet service	0	0	Action Required ***
Critical	Enforce Kubelet client TLS authentication	0	0	Action Required ***
High	Forbidden Container Registries	0	27	Action Required **
High	Resources memory limit and request	0	27	Action Required **
High	Applications credentials in configuration files	0	47	Action Required **
High	Resources CPU limit and request	0	27	Action Required **
High	Workloads with Critical vulnerabilities exposed to...	0	0	Action Required *
High	Workloads with RCE vulnerabilities exposed to exte...	0	0	Action Required *
Medium	Non-root containers	14	27	48%
Medium	Allow privilege escalation	14	27	48%
Medium	Ingress and Egress blocked	14	27	48%
Medium	Automatic mapping of service account	12	69	83%
Medium	CoreDNS poisoning	1	73	99%
Medium	Access container service account	6	48	88%
Medium	Cluster internal networking	1	5	80%
Medium	Linux hardening	14	27	48%
Medium	Configured liveness probe	3	27	89%
Medium	Secret/ETCD encryption enabled	1	1	0%
Medium	Audit logs enabled	1	1	0%
Medium	Images from allowed registry	0	27	Action Required **
Medium	Workloads with excessive amount of vulnerabilities	0	0	Action Required *
Medium	CVE-2022-0492-cgroups-container-escape	14	27	48%
Low	Immutable container filesystem	14	27	48%
Low	Configured readiness probe	3	27	89%
Low	Kubernetes CronJob	2	2	0%
Low	Network mapping	1	5	80%
Low	Pods in default namespace	12	27	56%
Low	PSP enabled	1	1	0%
Low	Image pull policy on latest tag	1	27	96%
Low	Label usage for resources	12	27	56%
Low	K8s common labels usage	12	27	56%
RESOURCE SUMMARY		22	203	70.13%

FRAMEWORKS: AllControls (compliance: 69.68), NSA (compliance: 62.64), MITRE (compliance: 72.82)

Figure 3: At the end of a Kubescape run, the tool outputs a complete overview of all the issues it identified. By default, the MITRE and NSA frameworks determine what constitutes an issue.

kubect1 package with your distribution's package manager.

## First Launch

Now you can start looking for security and compliance issues with Kubescape. The

```
kubescape scan --verbose
```

command starts an extensive scan that checks the target instance of K8s for all the rules it contains. As described, this includes MITRE and NSA frameworks, as well as some rules from Kubescape's own ruleset. Once the program has finished its work – and this can take quite a while, depending on the scope of the installation – it displays a table with the findings. However, interpreting them is not as easy for the untrained eye as you might expect or desire.

The table shown in [Figure 3](#) is initially divided into five columns. The first column is easy: It explains the severity of an issue or a deviation from the compliance regulations. Basically, if entries classified as *Critical* or *High* appear, increased vigilance and typically rapid intervention are the order of the day. Column 2 shows the name of the test performed; you need to develop a feel for working with Kubescape because it is often not clear at first glance whether an issue relates to compliance or security. At least if Kubescape detects a problem that has a CVE number, it usually uses this number in the *Control Name* column. Incidentally, the name of this column is derived by the names of individual checkpoints, which in the context of compliance certification, are typically referred to as controls [\[4\]](#).

Kubescape is exemplary in that most of the controls have descriptive names. *Disable anonymous access to Kubelet service*, for example, clearly informs you of the issue (anonymous access to the kubelet is permitted) and also provides appropriate instructions.

The next two columns, *Failed Resources* and *All Resources*, on the other hand, regularly prompt exasperation. However, fear not; whereas Kubescape lists the resources that it has checked in a

Kubernetes cluster in the *All Resources* column, the *Failed Resources* column lists the subset of the total number of services checked that Kubescape has identified as non-compliant in the context of a specific control entry.

The crux of the matter is that not every Kubescape control relates exclusively to resources in Kubernetes. It can happen that Kubescape finds critical errors but displays 0 in both columns, as in the example of anonymous access to the Kubelet service. These are not errors then that can be assigned to individual resources in K8s, but to either errors in components of the K8s infrastructure itself or meta checks. If too many tests for controls in the *Medium* category fail, for example, Kubescape displays a separate warning.

Finally, the last column records the percent compliance of an environment with regard to certain controls. Any control that displays a message (e.g., *Action Required*) instead of a value in this field, is an immediate prompt for you to take action. Otherwise, a value indicates how many of the tested resources in Kubernetes meet the specifications. Above the table Kubescape provides an overview and, below, a more detailed presentation of compliance fulfillment in terms of the applied rules (i.e., MITRE, NSA, or both). If you are not familiar with the Kubescape tabular output format, you can use the `--format` parameter to extend the output, including JSON, PDF, or HTML format:

```
kubescan scan --format json 2
--format-version v2 2
--output results.json2
kubescan scan --format pdf 2
--output results.pdf
kubescan scan --format html 2
--output results.html
```

If you want Kubescan to list the resources that have passed all tests, as well, add the `--verbose` parameter.

## Special Tasks

Kubescape also can be used outside a running Kubernetes cluster to check resources before they even find their

way into the cluster. For example, if you have a local folder full of YAML or JSON files that you want to transfer to Kubernetes later by running `kubect1 apply`, you can run either of

```
kubescape scan *.yaml
kubescape scan *.json
```

for a command-line check directly in the folder. The same approach works with a local folder full of Helm charts, which you can simply pass in to `kubescape scan` as a parameter.

## Conclusions

Kubescape is a powerful tool for efficiently monitoring running Kubernetes instances and resources that are not yet active in the cluster to help admins navigate the highly complex world of Kubernetes far more safely than would be the case without the tool. Because Kubernetes comprises so many layers and levels and potential add-ons, admins find it difficult to keep track of all the factors they need to check without an automatic helper in their toolbox. Kubescape is a massive help because the report shown at the end of a scan gives you a quick overview of what needs to be done. The software is available under a free license, is free of charge, and can be put into operation quickly, so it should be part of the standard repertoire of every company that uses Kubernetes or runs Kubernetes in its infrastructure. ■

### Info

- [1] Kubescape: [\[https://github.com/kubescape/kubescape\]](https://github.com/kubescape/kubescape)
- [2] Kubescape releases: [\[https://github.com/kubescape/packaging/releases\]](https://github.com/kubescape/packaging/releases)
- [3] Installing kubect1: [\[https://kubernetes.io/docs/tasks/tools/install-kubect1-linux/\]](https://kubernetes.io/docs/tasks/tools/install-kubect1-linux/)
- [4] Controls in Kubescape: [\[https://hub.armosec.io/docs/controls\]](https://hub.armosec.io/docs/controls)

### The Author

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



# Looking for Open Source talent?




Dedicated to professionals in the open source industry, Open Source JobHub will help you connect with qualified candidates.

OpenSource  
JOB HUB



[opensourcejobhub.com](https://opensourcejobhub.com)





Self-hosted remote support

# Direct Line

RustDesk supports self-hosted cross-platform remote support and maintenance. The client and optional basic server are open source and available free of charge. By Christian Knerrmann

**RustDesk software** [1] for remote support and maintenance is a recommended alternative to the established relay servers from commercial vendors. The relatively new project offers source code on the GitHub platform [2] under the GNU Affero General Public License version 3 (AGPLv3). In this article, I look at the free version's basic feature set.

## Relay Servers

Depending on the operating system, Microsoft Remote Desktop Protocol (RDP), Apple Remote Desktop (ARD), and Virtual Network Computing (VNC) have all proven their value as protocols for accessing the graphical user interface (GUI) of remote computers on local networks. However, in today's world, with remote work becoming increasingly common, remote maintenance software that communicates securely and reliably over public networks is becoming indispensable for companies.

Regardless of whether users are looking for help on private company networks, family and friends need assistance in the home office, or someone runs into problems while on the move on a mobile network, contacting

devices securely over the Internet is not easy. The established providers of remote maintenance software solve this dilemma by running publicly accessible relay servers.

Clients log on to the relay server, which typically assigns them a random identification number (ID) and establishes a secure connection by exchanging the ID and password. As a rule, however, the known players charge a monthly or annual fee and keep the source code of their software a secret. Besides the costs, you also need to trust the provider.

## Cross-Platform, Free, Open Source

Development of RustDesk started in 2020 and has gained significant momentum since 2023. According to GitHub, the main developer of the software – operating under the company name Purslane Ltd. – is based in Singapore, although supporters around the world contribute to the project. As the name suggests, the software is based on the Rust programming language. The optional server is also open source, at least with a basic feature set, and is available free of charge.

Like its competitors, Purslane also offers a commercial Pro version at two rates. The additional functions include access control, LDAP integration, and single sign-on (SSO) in the top-of-the-range version [3].

The client component supports Microsoft Windows, macOS, and Linux – macOS and Linux in conjunction with both x86-64 and ARM processors, whereas Windows currently only runs on the x86-64 architecture. RustDesk also supports the iOS and Android mobile operating systems.

Apple users will find the client in the App Store. Android users can use the Google Play Store or the alternative F-Droid app store, or they can pick up an Android package (APK) directly from the project site. However, the APK requires relaxed security settings on the end device to sideload apps.

For macOS and Android, the online documentation describes the authorizations required to allow remote access. iOS devices can currently only provide support, but cannot accept incoming remote sessions themselves. The mobile clients also offer useful functions for remote control of the keyboard and mouse, including touch.

Photo by Karsten Wüth on Unsplash

## Linux Restrictions

The client for Linux currently primarily supports the X Window System (X11) for incoming remote access; support for the Wayland display server is still experimental with limited usability. Users of Wayland, which has widely replaced X11 as the standard, can offer help, but can only allow remote maintenance if they actively confirm the connection request. The client expressly points out that unattended access only works in conjunction with X11.

The Android app cannot be used meaningfully in the special case of Google ChromeOS because it is only designed for small mobile device displays and touch operation. However, you can install the Debian package of the RustDesk client in the ChromeOS Linux development environment and at least use it to establish remote sessions to other systems. The web client source code is not available on GitHub because it is still reserved for the commercial Pro version of the server, so I will not look at it in this article. Before turning to the technology, I should mention that the project's online documentation is available in English and 10 other languages; unfortunately, the docs do not keep pace with the software's development. During testing, I discovered that various client and server functions and options are not described in the documentation or now behave differently. However, don't let this deter you. The basic functions of the RustDesk client and server are largely self-explanatory and practical.

## Server as the Switchboard

The server runs on Linux and Windows. I will address the various installation options later; suffice to say for now that you can either install directly or use containers for the Docker and Podman container managers. Under the hood, the server comprises two services: the relay server (hbbr) and the signal server (hbbs), which the project's

online documentation also refers to as the rendezvous server or simply the ID server.

A client connects to the signal server and receives a unique ID and password; another client can use this password to connect remotely. The clients then attempt to establish a direct connection with end-to-end encryption. If not possible, the relay server enters play, and the clients communicate through it indirectly. You do not need to install your own server to get started. The project currently operates its own free public server instances. With an earlier version, the manufacturer still indicated the number and locations of the servers in the online documentation, but unfortunately this is no longer the case. As with commercial providers, this means you need to trust the server operators. A dedicated server is recommended for production. If you just want to use RustDesk on an internal network, you can rely on direct IP access without a server, but in this case, the clients' communication is unencrypted. A server is therefore the better choice. Before turning your attention to your server, I'll take a look at the client functions.

## Windows Client

On Microsoft Windows, you have the choice of installing the RustDesk client or using a portable version. The project's website points users to the latest stable version of the client in the GitHub repository. The portable executable file opens the client directly but also offers the option of installing retroactively with a note saying that installation can help avoid potential issues with Windows User Account Control (UAC).

If you decide to install the program, the setup routine offers the option of changing the installation path and creating shortcuts in the Start menu and on the desktop by default; you can disable these if you like. A virtual display driver is also installed by default, which is RustDesk's way of enabling access to systems that do not have a connected monitor.

As soon as you start the client, it contacts the project's public servers and reports at the bottom of the window: *Ready, For faster connection, please set up your own server.* The link redirects you to the website with pricing information. Follow the link for the free plan for the installation instructions and the downloadable resources on GitHub.

## Helping and Finding Help by ID or IP Address

The interface is similar to other remote maintenance tools; even less experienced users will quickly find their way around. In the left half of the window, the client shows your local system's ID with the password for access by an external helper below it. You can let the client generate a new, random password or manually set your own permanent password in the settings. The client helps you choose a strong password and recommends a mix of upper- and lowercase letters and numbers with a length of eight or more characters.

The three vertical dots take you to the advanced settings, which, up to the earlier v1.1.9 client, still took the form of a context menu; from v1.2.x (used here) a separate tab has several subcategories. In the advanced settings, you can manage the authorizations for remote sessions in the *Security* category and enable direct IP access and store the coordinates of your own server under *Network*. If direct IP access is enabled, you can reach the client with its IP address instead of the RustDesk ID, provided a direct network connection to the person seeking help is possible.

In the main area of the window on the right, you can contact other RustDesk clients by entering the ID of the remote site in the Control Remote Desktop field and start either a file transfer or a graphical session by clicking *Connect*. In both cases, you only need the other party's password to accept the incoming connection request if no one is at the other end.

In the area below, RustDesk remembers your *Recent Sessions*, which you can remove or add to your personal *Favorites*. On the *Discovered* tab, the client displays other RustDesk clients on the same network segment, provided you have not disabled automatic detection in their settings. The *Address Book* tab next to it requires a login with a username and password. However, this is reserved for users of the commercial Pro version of the self-hosted server.

As soon as you start a connection, the person seeking help will see a pop-up window at the other end with your local username, your RustDesk ID, and the active authorizations. By default, this means the ability to use the keyboard and mouse, the clipboard, bidirectional file transfer, audio transmissions, TCP tunneling, remote restart, and session logging.

The person seeking help can reject or accept the request. If accepted, the connection is opened, even if no password is sent. The pop-up window remains active while the connection is live. A person seeking help can terminate the session at any time and revoke or grant individual authorizations on an ongoing basis.

## Versatile Options

Once the connection is established, the top bar in the window shows the ID at the other end. The icon to the top left shows the type of connection; you can see it in plain text by mousing over the icon. A green shield with a checkmark means a direct and encrypted connection (Figure 1). A green shield with a circular arrow indicates a mediated and encrypted connection routed by a relay server. A red shield with an X represents an unencrypted connection by direct IP access.

You can use the other buttons in the middle drop-down bar to toggle between window and full-screen mode, open text chat or voice call with the other party, and transfer files in either direction. RustDesk uses a separate window with a three-column layout to handle file transfers. Two of the columns show the local and remote drives, and the third column lists the completed transfers (Figure 2).

The client also supports a direct TCP tunnel, which you can use to map a port on your local machine to a port on the target machine. Further options include the ability to lock the computer on the remote side or to send a

Ctrl + Alt + Del keyboard shortcut to unlock the remote computer again.

However, the shortcut only works if the client on the remote computer is the installed version and not the portable version, because only the installed client supports this ability.

You can optionally block the remote computer for local user input and lift this block again. The screen icon influences the display and quality of the remote session. You can display the session at its original resolution or use scaling, with a choice between balanced compression, quality-optimized compression, or optimized response time. Depending on the available bandwidth, optimized response time can cause visible fragmentation. For remote control of unattended target systems, a useful option automatically locks the target at the end of the session, if so desired. You can also save the operating system password in RustDesk to unlock the target automatically when reconnecting.

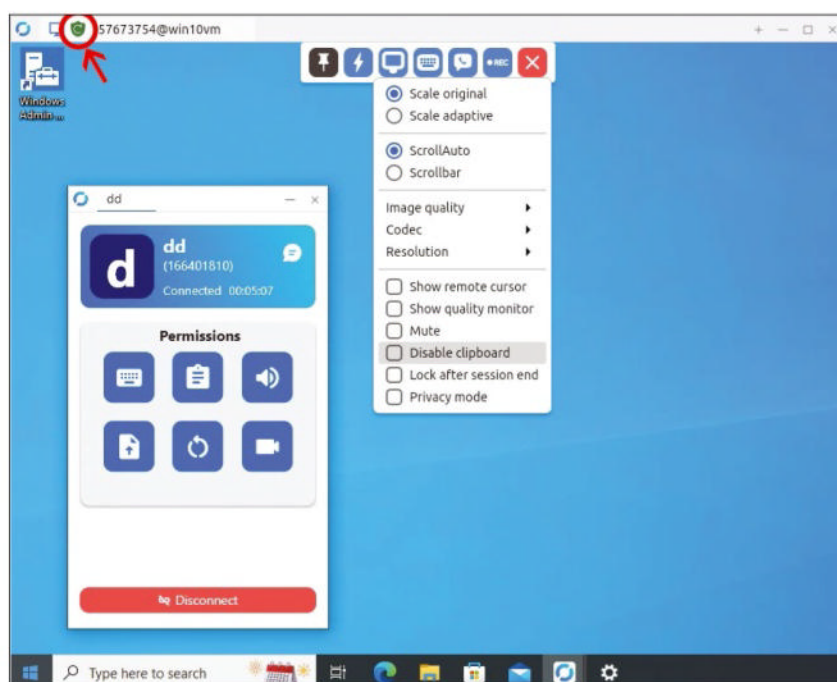
## Setting Up Your Server

RustDesk lets you operate on a self-hosted server; several variants come with or without a container manager [4]. Smaller companies and power users will be pleased to hear that the RustDesk server runs in combination with Docker on network-attached storage (NAS) systems by Synology. The online documentation guides you through the setup process, which consists of a few simple steps in the Synology Disk Station Manager (DSM) web interface without shell access. I tested my own instances on Microsoft Windows Server 2022 and Ubuntu 20.04 LTS (Focal Fossa). The steps for Linux should work on any distribution with Debian underpinnings.

## Getting Started with Linux

To begin, update the Linux system:

```
sudo apt update
sudo apt upgrade
```



**Figure 1:** The RustDesk client offers a wide range of options for configuring the display and quality of sessions.



The next step simply involves following the recommendations in the RustDesk documentation and enabling and configuring the firewall:

```
sudo ufw allow 22/tcp
sudo ufw allow 21115:21119/tcp
sudo ufw allow 21116/udp
sudo ufw allow 8080/tcp
sudo ufw enable
```

The first command makes sure you can use SSH for access, even if the firewall is active, and do not accidentally lock yourself out. The next three lines open up the ports required for the RustDesk server components. The TCP port 8000 of the minimalist Go HTTP File Server (gohttpserver) is optional. You only need this server if you want to use the automatically generated scripts for client installation; I will get back to this process in a moment. Because the web server uses unencrypted HTTP, I do not recommend publishing on the Internet; send the installation scripts to your clients by some other means. You need to allow the remaining ports on the other firewalls that protect your server, whether on your local network or cloud provider side. Now

download, make executable, and run the RustDesk installation script [5]:

```
wget https://raw.githubusercontent.com/techahold/rustdeskinstall/master/install.sh
chmod +x install.sh
./install.sh
```

In the first step, the script asks whether you want to install the server by its IP address or its DNS name. If you choose the DNS name, you need to enter it in the next step. The setup wizard then offers to set up the Go HTTP server.

Finally, the setup routine outputs the RustDesk server's public key and the admin user's password for accessing the web server. You will need to keep this information safe for later use. If you want to update your RustDesk server, the steps are the same as for the installation:

```
wget https://raw.githubusercontent.com/techahold/rustdeskinstall/master/update.sh
chmod +x update.sh
./update.sh
```

As an alternative to the script, the online documentation describes manual

installation by Docker Compose or without Docker by the PM2 process manager for Node.js applications.

## Installing the Server on Windows

For Windows, the documentation also guides you through the manual setup in conjunction with PM2 or the non-sucking service manager (NSSM). In our lab, I noted at this point that the software development was ahead of the documentation and that manual intervention is no longer needed on Windows.

Simply download the ZIP archive with the current setup package for Windows [6]. The setup routine installs the server components in C:\Program Files\RustDeskServer and drops shortcuts to the GUI onto the desktop and into the Start menu. Open the GUI and select *Services | Start* from the menubar. The server then automatically registers the hbbr and hbbs services on the system, sets their start type to *Automatic*, and starts the services immediately. When first launched, the RustDesk server generates a public-private key pair for client access and stores this

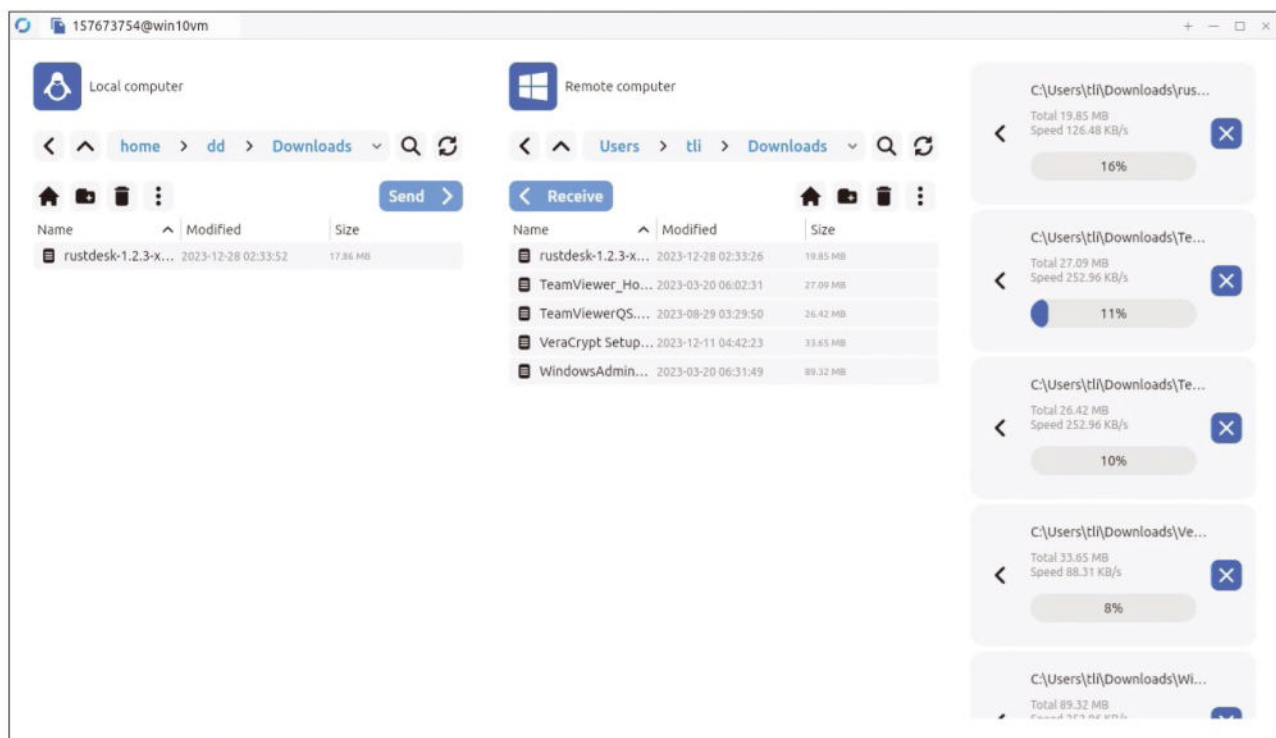


Figure 2: RustDesk displays bidirectional file transfers in a separate window.

in `C:\Program Files\RustDeskServer\bin`. You need to open the file with the PUB extension and save the string it contains to configure your clients. The server version (v1.1.8 at the time of testing) added incoming rules to the Windows firewall according to the RustDesk server's two program files. However, client access still wouldn't work. If you experience this, add incoming rules for the TCP and UDP ports mentioned previously. The server is now ready for connections, but how do the clients find their way to your server?

## Bringing the Client and Server Together

Call up the advanced settings in the client interface from the three vertical dots next to the ID. For clients up to v1.1.9, select the *ID/Relay Server* option from the context menu. As of v1.2.0, you will find the required options in the Network area of the Settings tab. If hbbs and hbbr are running on the same server, you just need to enter the IP address or DNS name of your server in the ID Server field. The server's public key in the Key field is also mandatory. As soon as you have accepted the settings, the status of the client in the footer of the window should change to *Ready* without anything else telling you that you now have your own server. If you want to avoid the manual overhead of distributing the address and key to all users and configuring the clients, RustDesk offers several approaches to simplifying the process.

The Linux server provides a shell script for Linux and a PowerShell script for Windows with the optional Go HTTP File Server on `http://<your-server>:8000` (Figure 3). Both automatically install the RustDesk client on the respective platforms with default settings that match your server. On Windows, you can alternatively include the server and key in the file name of the portable client, which, however, leads to very long file names in excess of 80 characters. For Linux, macOS, and the mobile platforms, you have no alternative to manual configuration.

## DIY Clients

Alternatively, you can compile the client software from the sources yourself. The developer documentation describes the procedure for setting up the required tools on various platforms. Without any local tools or programming knowledge, GitHub Actions [7] gives you your own client, provided you have a free GitHub account. To do this, create a fork of the RustDesk project in the GitHub web front end, enter the name or address of your server and its public key in the repository, then start a GitHub Action. The action automatically generates client packages that contain the server and public key of your server preconfigured. However, it should be noted that the process has to make do with limited resources and can take several hours. The fork is also publicly accessible. A private repository

limits the number of build processes per month, so you might have to switch to a commercial account. Even then, however, the fields for server and key in the client configuration are visible to end users in plain text. Any user who has this information can then use your server. Only the Pro version of the server restricts access by user names and passwords.

## Conclusions

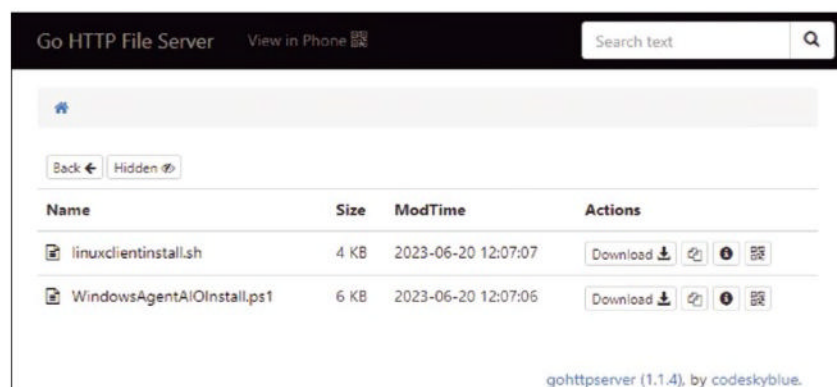
RustDesk turns out to be a powerful tool for solving technical problems remotely, which makes it a useful and free alternative to commercial tools for the same purpose. RustDesk is cross-platform software with support for Windows, macOS, Linux, and mobile devices. The client options leave virtually nothing to be desired in production, and thanks to the numerous options for operating your own server, you can keep the system completely under your control. On the downside, restricting access so that only authenticated and authorized users can use your publicly accessible server means signing up for one of the commercial Pro options. ■

### Info

- [1] RustDesk: [\[https://rustdesk.com\]](https://rustdesk.com)
- [2] RustDesk source code: [\[https://github.com/rustdesk/\]](https://github.com/rustdesk/)
- [3] Price models for Pro variants: [\[https://rustdesk.com/pricing.html\]](https://rustdesk.com/pricing.html)
- [4] Self-hosting RustDesk: [\[https://rustdesk.com/docs/en/self-host/\]](https://rustdesk.com/docs/en/self-host/)
- [5] Linux installation script: [\[https://github.com/techahold/rustdeskinstall/\]](https://github.com/techahold/rustdeskinstall/)
- [6] Setup package for Windows: [\[https://github.com/rustdesk/rustdesk-server/releases\]](https://github.com/rustdesk/rustdesk-server/releases)
- [7] GitHub Actions with RustDesk: [\[https://rustdesk.com/docs/en/dev/build/all/\]](https://rustdesk.com/docs/en/dev/build/all/)

### Author

Christian Knermann is Head of IT-Management at Fraunhofer UMSICHT, a German research institute. He's written freelance about computing technology since 2006.

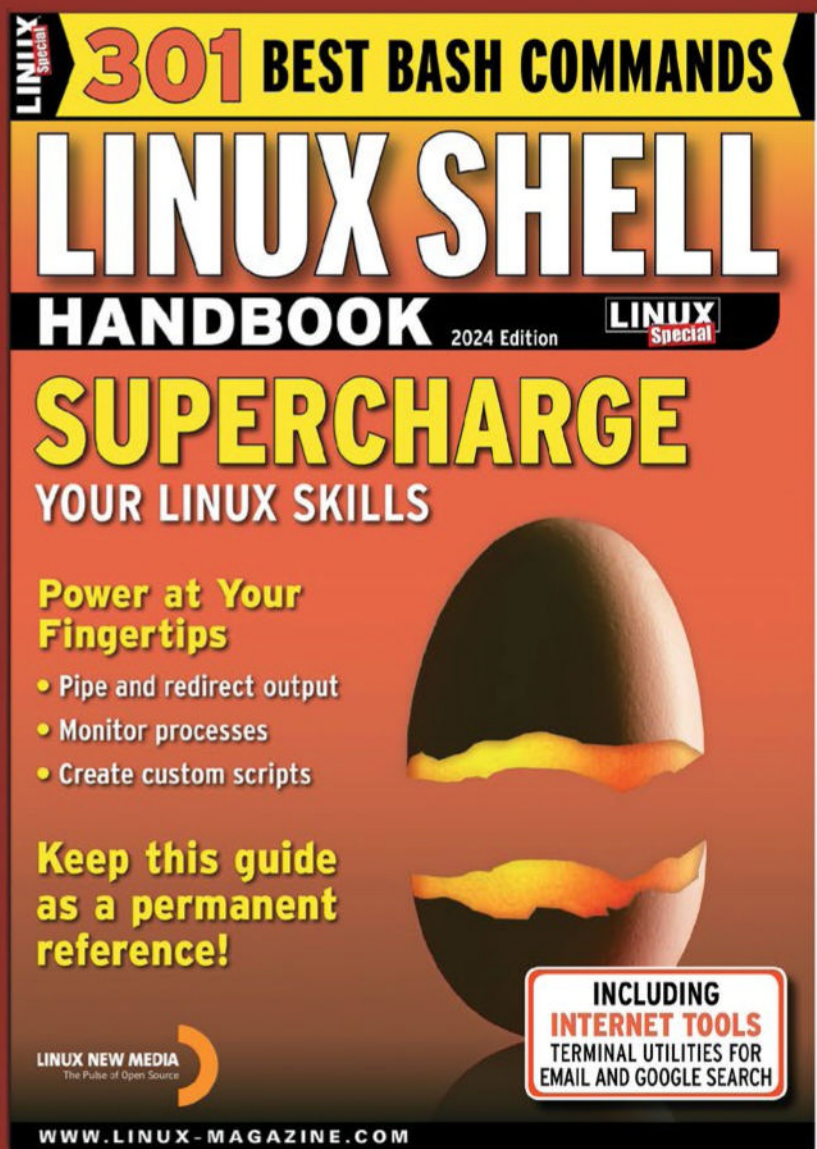


**Figure 3:** The RustDesk server on Linux simplifies the installation of clients with the use of preconfigured scripts.

# THINK LIKE THE EXPERTS

Linux Shell Handbook 2024 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
- Internet Tools
- And much more!



ORDER ONLINE:  
[shop.linuxnewmedia.com](http://shop.linuxnewmedia.com)





## Hyper-V failover with Azure Site Recovery

# Got Your Back

Back up Hyper-V with Azure Site Recovery and prepare for failover.

By Christian Schulenburg

In many cases corporate IT is set up as virtual environments, which makes it all the more important to have recovery strategies in place for the IT landscape in the event of a disaster. Recovery is all about strategies and actions that help restore IT systems after a catastrophic event – be it a natural disaster, a hardware failure, or a cyberattack. The aim is to minimize data loss and get the services up and running as quickly as possible.

Hyper-V is Microsoft's server virtualization product that lets you run multiple virtual machines (VMs) on a single physical host. Because VMs often host critical business applications and data, fast and reliable recovery after a disaster is crucial.

Microsoft Azure is a robust and scalable cloud platform ideal for disaster recovery scenarios, especially in combination with Hyper-V or other hypervisors. Azure Site Recovery (ASR) is Microsoft's disaster recovery as a service (DRaaS). Thanks to ASR, companies can replicate their Hyper-V VMs in Azure and restore them as needed, ensuring a seamless transition in the event of an emergency and enabling companies to ramp up their resources quickly in the cloud.

For this article, I used Hyper-V to set up a virtual server on Windows Server 2019 and replicated it with an Azure subscription. An existing Azure environment was used in this case; however, you can also take out a free Azure subscription [1] and get started with an initial credit of \$200.

### Preparing the Azure Environment

Setting up ASR for Hyper-V is a multistep process and starts on the Azure side. The login account in Azure must be able to create a VM in the selected resource group and on the selected virtual network. To do this, the account requires the Virtual Machine Contributor and Site Recovery Contributor roles to manage the recovery processes. You should work as the environment's standard administrator, which means you automatically have all the required authorizations. To prepare Azure, you first create a separate storage account, a resource group, the recovery services vault, and a virtual network. You also need a storage account in Azure to store the VM images. To set this up, select *Create a resource* on the Azure start

screen and select *Storage accounts* | *Create* (Figure 1). Besides the subscription, name, and region, you also need to select a resource group, which involves creating a new group in which to create all the other services. The *Geo-redundant storage (GRS)* option ensures redundancy. Unlike redundancies in the primary region, this option asynchronously copies the data to a second region hundreds of kilometers away from the primary region.

In the next step, you create the recovery services vault containing the metadata and configuration information for the VMs. The service is launched from the *Recovery Services vaults* entry on the Azure start page, where you need to re-enter the subscription, region, resource group, and name for the vault. The vault sets up quickly. Carry on by using the *Virtual networks* item to create a network environment for the landscape; the VMs will be linked to this network after a failover.

In addition to the standard parameters such as name and resource groups, you also need to define the network area in which the VM will run. In this example, the local network and the virtual network in Azure are not linked, but a connection with Azure virtual private network (VPN) can be set up so that clients can continue to work seamlessly

Lead Image © crazymedia, 123RF.com

at the local site. The preparations for Azure are now complete for the time being; the next step is to prepare the local Hyper-V server.

## Setting Up a Local Agent

Hyper-V Server can be used on Windows Server 2012 R2 or newer. All

operating systems that are supported by Azure and can be executed there can be used as guest operating systems for replication. Additionally, the Hyper-V server requires an Internet connection to various resources for replication. The resources include \*.hypervrecoverymanager.windows-azure.com for replication and \*.blob.core.windows.net for storage. The only port used here is HTTPS (443). Information on the websites and the extended requirements can be found online [2].

In addition to Internet access, you must ensure that remote access is enabled for the VMs so you can access the servers in Azure after a failover. I enabled Remote Desktop Protocol (RDP) for this on my Windows test server, which is all that needs to be considered in terms of the local environment.

The next step is to prepare the Azure infrastructure before setting up disaster recovery for the local Hyper-V VMs on the Azure side. To do this, navigate to your Recovery Services vault in Azure. In the *Site Recovery* sidebar item, select the *Prepare infrastructure* option under the *Hyper-V machines to Azure* tile (Figure 2).

In the wizard, you first choose to complete deployment planning later.

**Figure 1:** As a prerequisite, various resources need to be set up in Azure, including a storage account for storing the VM images.

**Figure 2:** Once the preparations are complete in Azure and the local environment, they need to be configured through the *Prepare infrastructure* link.

The Azure Site Recovery Deployment Planner basically helps you determine the required bandwidth and Azure storage, which is not relevant in the context of this exercise. You can find a detailed article on the tool in the Azure documentation. In the next step, I chose not to use System Center Virtual Machine Manager (VMM) to manage my Hyper-V hosts, which means that I now have to enter a Hyper-V site. Because it has not been set up yet, I create a new one named *itadministratorVMVault*.

Up to this point, the local Hyper-V host is not known in Azure. The *Create Hyper-V site* entry opens a small window that can be used to download the *Microsoft Azure Site Recovery provider* and the vault registration key. I ran the installation wizard locally and integrated the registration. The setup involves just a few steps, and in future the *Microsoft Azure Site Recovery Service* will take care of replication and other tasks. During the installation in Azure, the server is added by selecting *Site Recovery Infrastructure | Hyper-V Hosts*. It could take some time for the newly added servers to become available.

Now you can enter the Hyper-V host in the source settings of the setup wizard. The target settings are covered later in the wizard. Select the subscription and the delivery

model. I used *Resource Manager* as the model, which is the standard model; the alternative is legacy provisioning. To simplify deployment and management, Microsoft recommends *Resource Manager* for all new resources. Next, create a new policy from the *Replication policy* tab (Figure 3) to define the settings for the retention history of recovery points and the frequency of application-consistent snapshots.

In the policy, I defined *Hyper-V* as the source type and *Azure* as the target type; the copy frequency was set to *5 Minutes*, the retention policy for recovery points to 2 hours, and the frequency of app-consistent snapshots to 1 hour. The replication start time chosen was *Immediately*; pressing *OK* confirmed the new policy. In the next step of the setup wizard, the overall configuration is displayed again and I selected *Create* to finish. Once the infrastructure preparations are complete, replication of the virtual computers can begin from the *Enable replication* entry in the *Site Recovery* menu item. You will see six tabs. On the first tab, select the Hyper-V source site created earlier (*itadministratorVMVault*). Under *Target environment*, the subscription, resource groups, provisioning model after a failover, and storage account are queried again; these settings can

be adjusted later in the replicated VM in Azure.

In the next step, you specify the VM from the local environment that you want to replicate. I selected the local test server here. The data carrier details then need to be entered under the *Replication settings* tab. The next step involves selecting a replication policy before completing the setup by clicking *Enable replication*. The setup is now complete, and the server is replicated in the Azure environment. The first replication takes some time, and the different states are displayed in Azure.

## Monitoring Replication

The recovery service dashboard shows all the monitoring information for site recovery in one place. The *Replicated items* section provides details of the integrity of all the computers in the vault for which replication is activated, so you can quickly discover whether you have any current problems with replication. The failover integrity is initially set to *Warning* after the first replication because a failover test has not yet been performed. (I look at this test later.) Errors relating to the configuration, subscription, or resources are displayed directly in the *Configuration Issues* section. In the lower

**Figure 3:** In addition to the source and target information, a configuration wizard sets up the replication policy, which defines recovery points, among other things.



area, the infrastructure view visualizes the infrastructure components involved in replication and the connectivity between servers and Azure services.

Details of the replicated elements can be found in the menu on the right or displayed by following the *View all* link. In the table view, columns can be added and removed and entries can be filtered; clicking on a computer reveals more details, including replication information, the recovery point objective (RPO), recovery points, failover readiness, errors, and events.

The recovery time objective (RTO) and RPO key performance indicators (KPIs) are important points of contingency management. The RPO refers to the period during which data loss occurs. The RTO defines how long it can take to restore normal operation after a failure. The times determined by the failover tests in Azure must be brought into line with your requirements. Times can be further reduced with Azure Traffic Manager.

The status of a replication can be tracked in Hyper-V Manager on the local server, as well, but with no local configuration options; actions cannot be managed locally. To receive information in the event of an error, email notification can be enabled in the *Alerts* section of the Azure vault.

## Testing the Failover

Various actions can be performed in the *Overview* of a backed-up machine, including *Test Failover*, *Cleanup test failover*, *Commit*, *Resynchronize*, and *Change recovery point* (Figure 4).

Certain actions, such as *Cleanup test failover*, can only be carried out for a specific status of the replicated element. The most important actions are the failover scenarios.

First, I'll take a look at the *Test Failover* option and run it in my environment. Testing a failover creates a copy of the backed up VM in Azure without affecting running replication processes or the production environment. This machine can be accessed and managed in the same way as any other virtual computer in Azure. A network must be selected for the failover during the test recovery. This network must be kept separate and have no connection to production so that the restored system can be tested without interfering.

Because I don't have any connections between the networks in this test, I used the standard network in Azure and was able to use RDP to connect to the server after the recovery. The production server kept running the whole time. At the end of the test, the *Cleanup test failover* action automatically removed all

VMs that were created in Azure during the exercise.

As the name suggests, *Planned failover* is used for planned outages. The local VM is then shut down, the latest data is synchronized, and the VM is started in Azure. No data is lost, but some downtime must be planned. After initializing and executing the failover, the VM must be tested. A *Commit* is performed for confirmation, and it deletes all the recovery points in Azure. The VM is now running completely in Azure. To achieve a failback to the local environment, you just need a planned failover in the other direction. I selected the local environment as the target. While the failover from Azure to the local site is running, Site Recovery shuts down the Azure VM, transfers the data to the local VM, and starts it. After the successful failback and checking the machines, another *Commit* is required for the failover in Azure, which deletes the Azure VM.

A failover usually occurs if you experience an unplanned outage or the primary site is not available. During the initialization step, you can specify whether you also want to shut down the local VM and replicate the data. Of course, this decision depends on whether the location and the VM are still accessible. If the option to shut down the

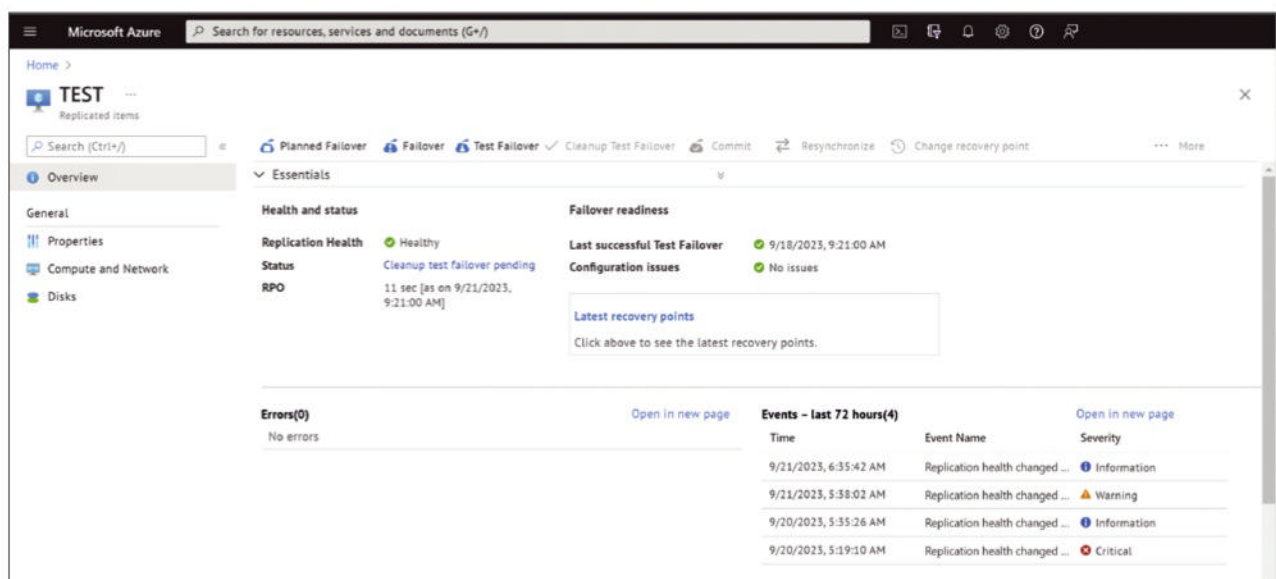


Figure 4: The status can be checked and various actions, such as a failover, can be triggered in the overview of a replicated element.

VM is not activated or Site Recovery cannot shut down the VM, the last recovery point in Azure is used for the restore. The failover takes place in any case, even if the VM cannot be shut down. Finally, another commit occurs, following which the VM is then ready for use. The failback procedure is the same as the procedure for a planned failover.

## Setting Up Restore Plans

In addition to the recovery of individual elements, several of them can be combined in recovery plans. A recovery plan defines how a failover is executed for several servers at once. The order specifies how the servers are started after the failover, which means that application landscapes can be displayed, and, for example, a database server can be started before an application server. The recovery plans can be used for failover and failback. A recovery plan groups machines together. Servers in the same group start in parallel, whereas the groups start one after the other.

Failover of local virtual servers to Azure is no problem. However, this route is not recommended. Instead

of the ASR service, Azure Migrate is the preferred approach to migrating VMs to Azure. If ASR is already in use, a failover can be finished with the *Complete Migration* action instead of with *Commit*. Further information on Azure Migrate can be found online [3].

## Cost Overview

A secure Hyper-V instance costs around \$25 per month. Billing is monthly on the basis of the daily average figures for the protected instances. Infrastructure costs also come into play, including memory usage, data transfer, and potentially a VPN connection. After a failover, the costs of the VM are added. You can use the Azure price calculator [4] to specify all Azure services used to obtain a cost estimate.

## Conclusions

Disaster recovery for Hyper-V environments always requires careful planning. Integrating Azure offers an easy approach to making local Hyper-V more resilient. The setup is very quick and easy. Don't forget to carry out regular tests so that you

can react quickly and efficiently in the event of a disaster. ASR itself reminds you of this advice, and the environment is constantly monitored. ASR does not replace regular backups, but it does give you a simple way to protect your environment against data loss. ■

---

### Info

- [1] Free Azure subscription: [<https://azure.microsoft.com/en-us/free>]
- [2] Preparing Hyper-V: [<https://learn.microsoft.com/en-us/azure/site-recovery/hyper-v-prepare-on-premises-tutorial>]
- [3] Azure Migrate: [<https://learn.microsoft.com/en-us/azure/migrate/migrate-services-overview>]
- [4] Azure pricing calculator: [<https://azure.microsoft.com/en-us/pricing/calculator/>]

---

### The Authors

Christian Schulenburg has been working in IT for more than 20 years and is a long-time Exchange MVP. He has been an IT specialist for systems integration and an IT consultant. He is currently working as a consultant for digitization at the Mecklenburg-Western Pomerania district council in Schwerin Germany. His main areas of activity are Salesforce and Microsoft technologies, with a particular focus on Exchange.

# Linux Magazine Subscription

Print and digital options  
12 issues per year



► SUBSCRIBE  
[shop.linuxnewmedia.com](http://shop.linuxnewmedia.com)

Expand your Linux skills:

- 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

Go farther and do more with Linux, subscribe today and never miss another issue!



Follow us



@linux\_pro



Linux Magazine



@linuxpromagazine



@linuxmagazine

## Need more Linux?

Subscribe free to Linux Update

Our free Linux Update newsletter delivers insightful articles and tech tips to your inbox every week.

[bit.ly/Linux-Update](http://bit.ly/Linux-Update)







Bottom: The new top

# From Top to Bottom

Bottom is the latest process and system monitoring terminal user interface tool, delivering lightweight but beautiful monitoring. By Federico Lucifredi

**Bottom (btm)** [1] is the humorously named successor to top. Blending mouse interaction and keystrokes in a terminal window, Bottom is pushing forward a crowded, ever-advancing field led until recently by **bashtop** [2] [3], its more performant cousin **bpytop** [4], and many others [5] alongside the evergreen original **top** [6]. Rust-based, Bottom promises a more lightweight and highly customizable cross-platform experience; on macOS, install via Brew (`brew install bottom`). On Ubuntu, the choice of hosting packages in GitHub repos instead of

Ubuntu Universe makes the command choice dependent on architecture in the apt path, but a straightforward Snap install is available:

```
sudo snap install btop; 2
sudo snap connect btop:removable-media
```

Refer to the project homepage for more details [7].

## Kicking Tires

Among so many options, what makes **btm** special? It provides immediate and

time series monitoring of CPU, memory, network activity, processes, and disk usage like other tools, but it does it with characteristic flair. An interesting example is found in the six pre-built color themes, their styles ranging from ice blue to retro-groovy cool. The *default* color scheme is selected when no selection is made, but you can see a few of the alternative choices available in **Figure 1**. Nothing should stop you from creating your own template if you do not like those supplied. Naturally the tool ships with its own man page, which comes in very

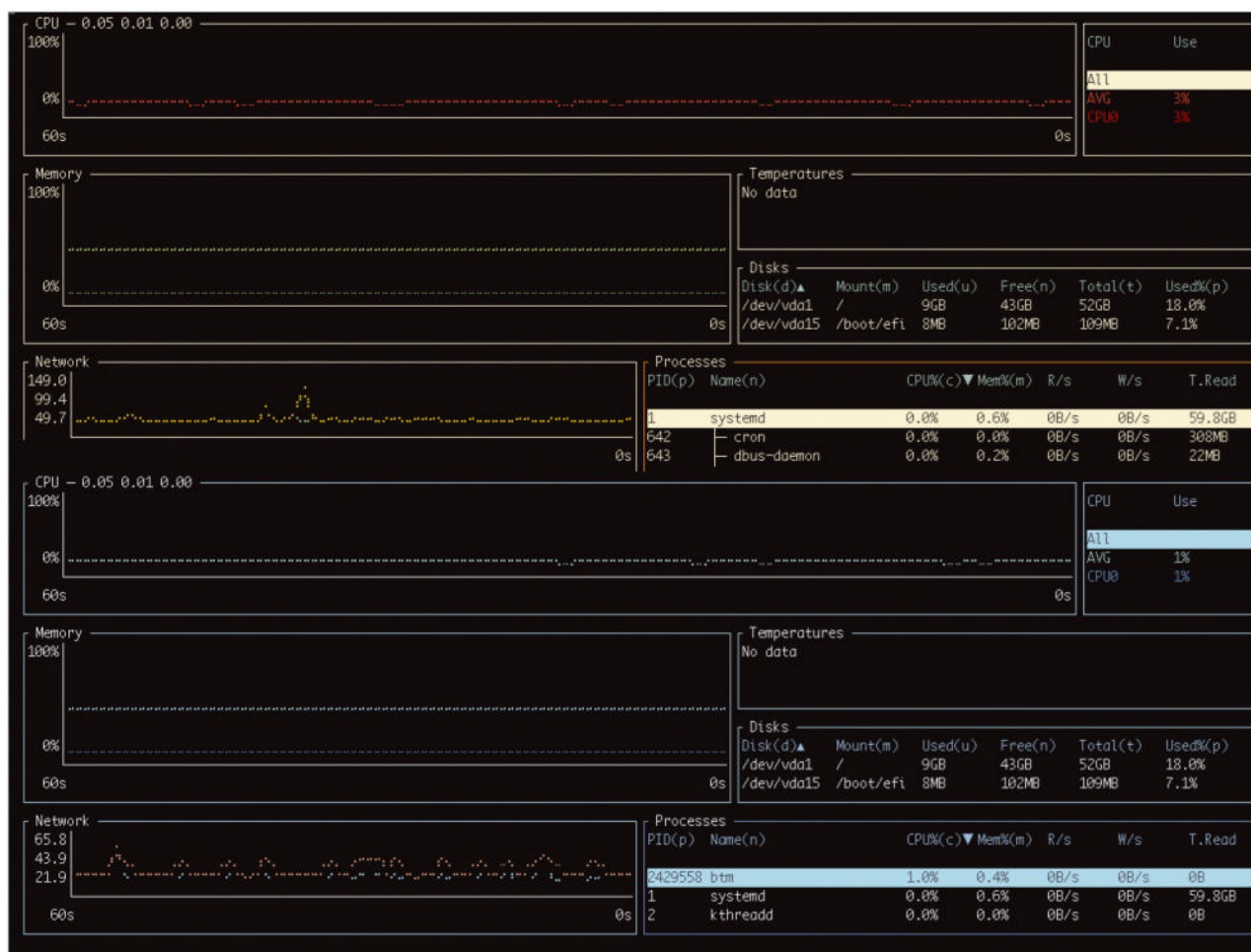


Figure 1: Up top, the gruvbox color scheme; at bottom, the nord color scheme.

Lead Image © Lucy Baldwin, 123RF.com

handy when studying command-line options; however, in interactive mode, all one really needs to remember is the ? keystroke. A handy in-tool pop-up several pages long

comprehensively details how to interact with each widget and its respective options, in what really ought to be a terminal user interface (TUI) best practice (Figure 2).

Fans of htop [8] point out it is trivial to reproduce the same visualization in Bottom with

```
btm -b -r 2000
```

(Figure 3), which can be handy as htop undergoes maintainer transition. Another interesting option is the process tree view, triggered by *t* or F5 (Figure 4), with the process widget at full screen (select with mouse click, press *e*). Also of interest is the possibility to view the full command instead of the binary's name (*P*). The F9 key, perhaps more easily remembered by *vi* users with the alternative mnemonic *dd*, is used to kill the selected process.

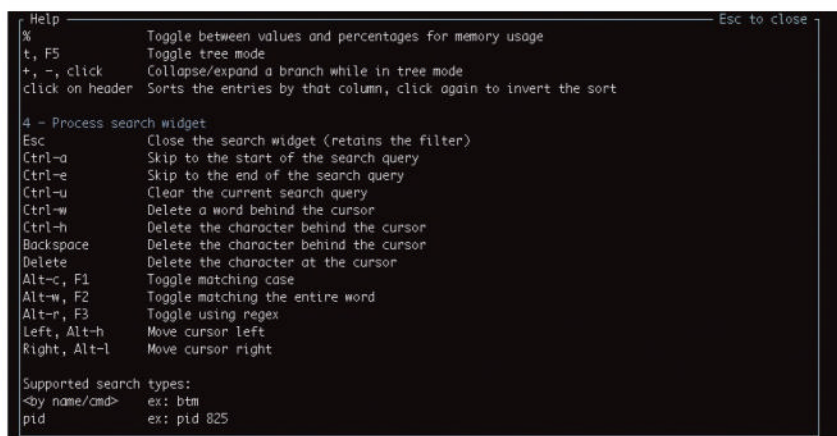


Figure 2: The application's help screen is always one keystroke away.

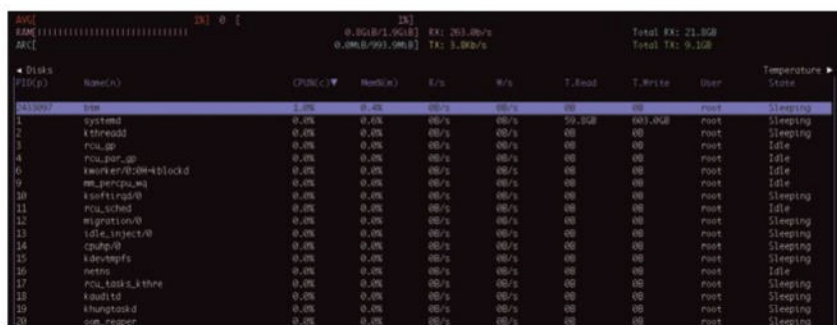


Figure 3: Process view in the style of htop, but rendered in Bottom.

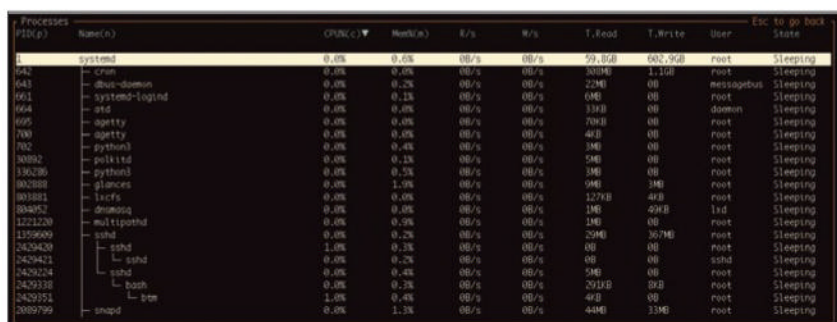


Figure 4: Process tree view with a color scheme applied.

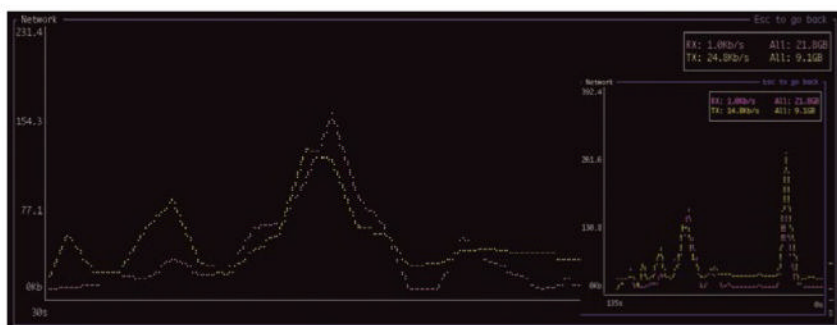


Figure 5: Network monitoring with a 30sec time base, and (inset) the same data compressed with a 135sec time axis.

## Searching and Sorting

Bottom includes extensive facilities to search, and hopefully find, processes and their command options. The / key starts the search, which can use just a string, or a full regular expression if that is required by the task. Sorting is most easily accomplished with a mouse click on the field of interest, in the process view, disk status, or temperatures widget. Time series widgets also respond to the scrollbar and equivalent touchpad gestures (+ or - will produce the same outcome) by compressing or expanding the time base represented on the x-axis. Figure 5 demonstrates the result of this action on the network monitoring widget, created with the assistance of the update freeze (*f*) key.

## Off to the Races

In a somewhat informal test, I have run *btm* in an otherwise completely idle Ubuntu Focal (20.04) single-core virtual machine hosted on Digital Ocean. With the benefit of the tool measuring itself in regards to system load for a few minutes, I observed it consistently hovering around 1% of CPU load (Figure 1, bottom view). In comparison, the much heftier glances [9], another favorite of mine, taxed the system between 2.5% and 3.4% in the same test setup (Figure 6). Since

glances is a more complete and complex tool, a fairer comparison is with bashtop, the tool that launched the current wave of TUI tool development. Highly variable, bashtop ran most of the time between 3.5% and 6% of CPU – and I watched it spike above 7% (Figure 7)!

Seven percent of CPU allocated to the measurement tool is clearly a lot. Thankfully, the author of bashtop iterated repeatedly on his subject, releasing bpytop first, which demonstrated a much reduced load of 1.2%-1.5% in this test (Figure 8) thanks to its Python redesign. Further improvement

is being demonstrated by new developments in btop [10], rewritten fully in C++ and the champion of this little test, hovering around a remarkably low 0.5% (Figure 9).

The final choice is yours, but clearly btop and btm are the current top contestants when it comes to keeping tooling interference in your measurements in check – and saving CPUs so you can forget them while they run on your system.

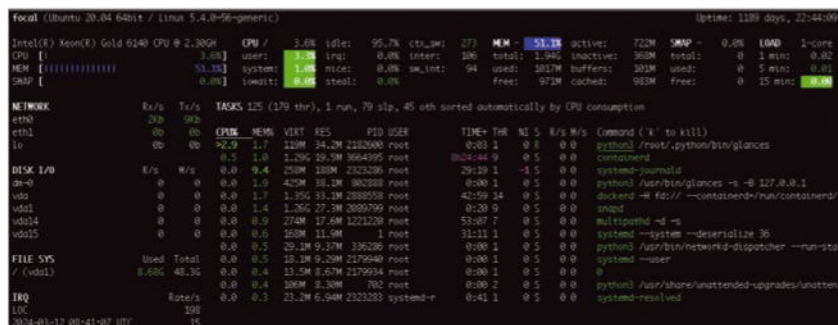


Figure 6: While somewhat long in the tooth, glances is one of the best terminal monitors around.

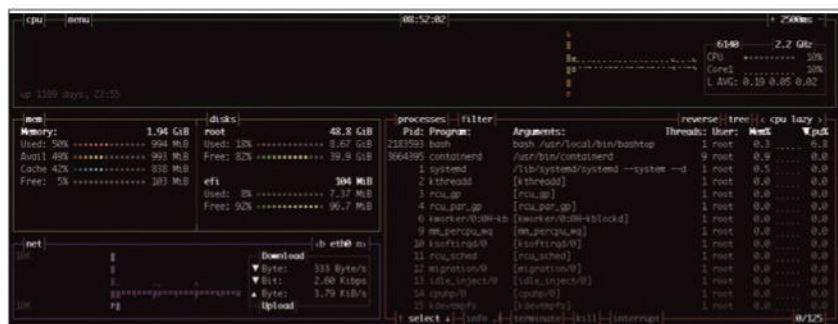


Figure 7: bashtop at work in my single-core test run.

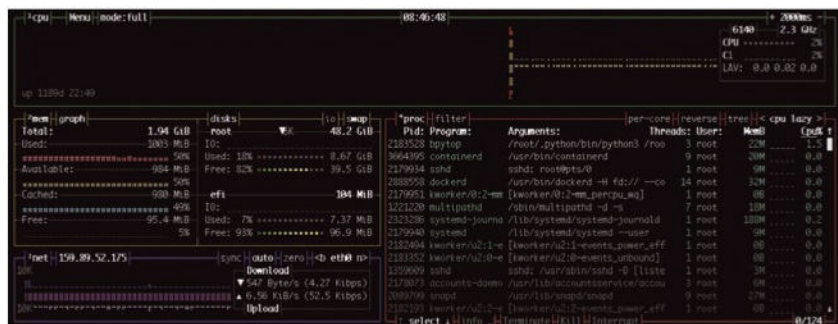


Figure 8: bpytop looking the same as Bashtop, but 75% less hungry for CPU cycles.

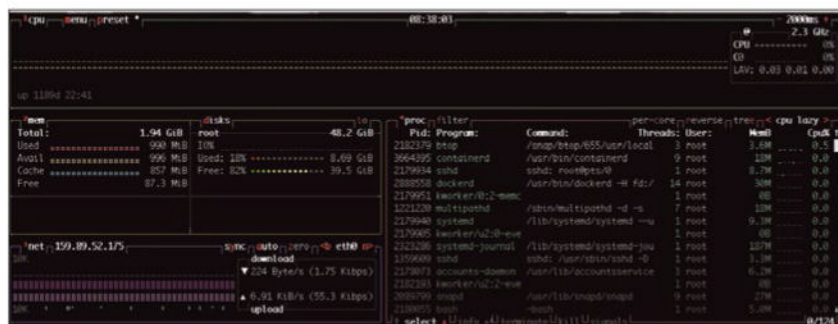


Figure 9: Thanks to its C++ implementation, btop is the new king of the hill.

## Info

- [1] Clement Tsang, Bottom: <https://github.com/ClementTsang/bottom>
- [2] Jakob P. Lilienberg, bashtop: <https://github.com/aristocratos/bashtop>
- [3] “Next-Generation Terminal UI Tools” by Federico Lucifredi, *ADMIN*, issue 64, 2021, <https://www.admin-magazine.com/Archive/2021/64/Next-generation-terminal-UI-tools>
- [4] Jakob P. Lilienberg, bpytop: <https://github.com/aristocratos/bpytop>
- [5] “Network Performance In-Terminal Graphics Tools” by Federico Lucifredi, *ADMIN*, issue 51, 2019, <https://www.admin-magazine.com/Archive/2019/51/Network-performance-in-terminal-graphics-tools>
- [6] “Exploring the Most Famous Performance Tool” by Federico Lucifredi, *ADMIN*, issue 46, 2018, <https://www.admin-magazine.com/Archive/2018/46/Exploring-the-most-famous-performance-tool>
- [7] Bottom, install and support: <https://github.com/ClementTsang/bottom#installation>
- [8] Hisham Muhammad, http: <https://en.wikipedia.org/wiki/Htop>
- [9] Nicolas Hennion, glances, <https://github.com/nicolargo/glances>
- [10] Jakob P. Lilienberg, btop: <https://github.com/aristocratos/btop>

## The Author

Federico Lucifredi (@0xf2) is the Product Management Director for Ceph Storage at IBM and Red Hat, formerly the Ubuntu Server Product Manager at Canonical, and the Linux “Systems Management Czar” at SUSE. He enjoys arcane hardware issues and shell-scripting mysteries and takes his McFlurry shaken, not stirred. You can read more from him in the new O'Reilly title *AWS System Administration*.



# ADMIN

Network & Security

## NEWSSTAND

Order online:  
<https://bit.ly/ADMIN-library>

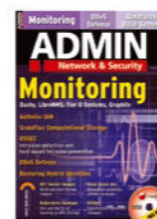
ADMIN is your source for technical solutions to real-world problems. Every issue is packed with practical articles on the topics you need, such as: security, cloud computing, DevOps, HPC, storage, and more! Explore our full catalog of back issues for specific topics or to complete your collection.

### #79 - January/February 2024

#### Monitoring

This issue takes a deep dive into monitoring solutions for your IT infrastructure, including Dashy, LibreNMS, Tier 0 systems, and Graphite.

On the DVD: FreeBSD 14.0



### #78 - November/December 2023

#### Domain-Driven Design

Business experts and developers collaborate to define domain models and business patterns that guide software development.

On the DVD: Fedora Server 39

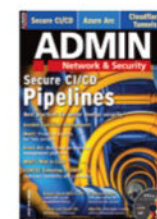


### #77 - September/October 2023

#### Secure CI/CD Pipelines

DevSecOps blends security into every step of the software development cycle.

On the DVD: IPFire 2.27

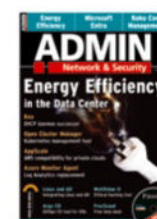


### #76 - July/August 2023

#### Energy Efficiency

The storage share of the total data center energy budget is expected to double by 2030, calling for more effective resource utilization.

On the DVD: Finnix 125 (Live boot)



### #75 - May/June 2023

#### Teamwork

Groupware, collaboration frameworks, chat servers, and a web app package manager allow your teams to exchange knowledge and collaborate on projects in a secure environment.

On the DVD: Ubuntu 23.04 "Lunar Lobster" Server Edition

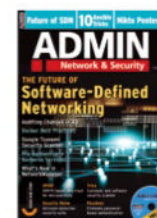


### #74 - March/April 2023

#### The Future of Software-Defined Networking

New projects out of the Open Networking Foundation provide a glimpse into the 5G network future, most likely software based and independent of proprietary hardware.

On the DVD: Kali Linux 2022.4



# 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](mailto:edit@admin-magazine.com).



## Authors

Amber Ankerholz	6
Nigel Bahadur	56
Prof. Harald Baier	10
Erik Bärwaldt	28
David Berube	40
Mark Heitbrink	70
Ken Hess	3
Christian Knemann	82
Martin Gerhard Loschwitz	16, 22, 76
Federico Lucifredi	94
Michael Mundt	10
Christian Schulenburg	88
Constantine Söldner	32
Guido Söldner	32, 66
Andreas Stolzenberger	36, 46
Nico Thiemer	52

## Contact Info

### Editor in Chief

Joe Casad, [jcasad@linuxnewmedia.com](mailto:jcasad@linuxnewmedia.com)

### Managing Editors

Rita L Sooby, [rsooby@linuxnewmedia.com](mailto:rsooby@linuxnewmedia.com)  
Lori White, [lwhite@linuxnewmedia.com](mailto: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

Lori White, Illustration based on graphics by  
[liudmilachernetska](mailto:liudmilachernetska), 123RF.com

### Advertising

Brian Osborn, [bosborn@linuxnewmedia.com](mailto:bosborn@linuxnewmedia.com)  
phone +49 8093 7779420

### Publisher

Brian Osborn

### Marketing Communications

Gwen Clark, [gclark@linuxnewmedia.com](mailto: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](mailto:cs@linuxnewmedia.com)  
Phone: 1-866-247-2802  
(Toll Free from the US and Canada)

For all other countries:  
Email: [subs@linuxnewmedia.com](mailto:subs@linuxnewmedia.com)  
[www.admin-magazine.com](http://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 © 2024 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 Kolibri Druck. Distributed by Seymour Distribution Ltd, United Kingdom

ADMIN (Print ISSN: 2045-0702, Online ISSN: 2831-9583, USPS No: 347-931) is published bimonthly by Linux New Media USA, LLC, and distributed in the USA by Asendia USA, 701 Ashland Ave, Folcroft PA. March/April 2024. Application to Mail at Periodicals Postage Prices is pending at Philadelphia, PA and additional mailing offices. POSTMASTER: send address changes to Linux Magazine, 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.



# Expose Rootkits with Invary Runtime Integrity.

Rootkits are designed to be invisible, undermining the very fabric of security solutions. With **Invary Runtime Integrity**, rootkits can no longer hide.

Invary is a force multiplier for your existing threat detection arsenal. The integrity of your operating system isn't just a priority; it's the battleground.

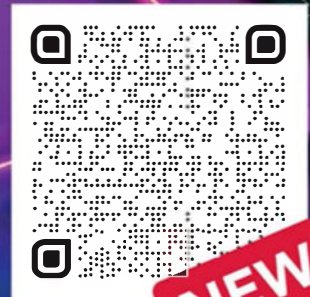


[invary.com](https://invary.com)



# HETZNER

# MULTITASKING AT ITS FINEST



**NEW**

## OUR NEW DEDICATED SERVER **AX162 WITH AMD EPYC™ 9454P**

The perfect match for your simultaneous workload - thanks to 48 CPU cores with 96 threads and up to 1152 GB DDR5 RAM

### DEDICATED SERVER AX162-R

- ✓ AMD EPYC™ 9454P  
48-Core, Genoa (Zen4)
- ✓ 8 x 32 GB DDR5 RDIMM
- ✓ 2 x 1,92 TB NVMe SSD
- ✓ Unlimited traffic
- ✓ Location Germany & Finland
- ✓ No minimum contract
- ✓ Setup fee € 79.00

monthly from **€ 199.00**  
\$ 215.24

### DEDICATED SERVER AX162-S

- ✓ AMD EPYC™ 9454P  
48-Core, Genoa (Zen4)
- ✓ 4 x 32 GB DDR5 RDIMM
- ✓ 2 x 3,84 TB NVMe SSD
- ✓ Unlimited traffic
- ✓ Location Germany & Finland
- ✓ No minimum contract
- ✓ Setup fee € 79.00

monthly from **€ 199.00**  
\$ 215.24

All prices exclude VAT and are subject to the terms and conditions of Hetzner Online GmbH. Prices are subject to change. All rights reserved by the respective manufacturers.

**www.hetzner.com**