

ADMIN

Network & Security

ISSUE 75

COLLABORATION AND COMMUNICATION

Teamwork

Cyn.in

Open source groupware

Matrix Open StandardSecure, decentralized
communication**Archiving in
Microsoft Teams****Sandstorm**

Self-hosted secure collaboration

Exchange Server Through 2025LINUX NEW MEDIA
The Pulse of Open Source**AWS Well-Architected
Framework**

Best practices in cloud design

Secured-Core Server6 new security features for
Win Server 2022**Azure Sphere**IoT with a focus
on security**Gophish**Testing for
phishing attacks

Attention Newsstand Readers

Welcome to the latest issue of *ADMIN*!

We hope you enjoy this month's selection of technical articles for IT professionals.

You may have noticed an increase in the cover price this month. Unfortunately, the rising costs of selling *ADMIN* on the newsstand require that we raise our prices.

You can save money and get *ADMIN* faster if you buy direct from us! You will always get the best price, and your direct support lets us continue to produce the quality content you expect from *ADMIN*.



US/Canada Customers
shop.linuxnewmedia.com



Customers in All Other Countries
sparkhaus-shop.com

Thank you for your continued support!



Happily Ever After Work

The quest for job satisfaction.

I have a small group of former co-worker friends that still maintain contact. However, our geographic diversity makes it impossible to gather at a local restaurant for those occasional intellectual exchanges. The other evening, we had a conference call to catch up on our jobs, families, vacations, and miscellaneous topics. One of the job-related topics we touched on was how our expectations of a job and job experience are often quite different. We decided we needed to modulate our expectations or sharpen our job choosers. Fixing one or the other would result in huge improvements in job satisfaction.

Job satisfaction, to some, might seem to be an oxymoron, but it shouldn't be. Job satisfaction should be more than just a phrase used by journalists and HR people; it should be a thing that's sought after by employees and promoted by company executives. I know it's a time-honored tradition to hate your job because they pay you to do it, but we spend one-third (or more if you're a system administrator) of our life at work, and we shouldn't hate one-third of our life experience.

Job satisfaction leads to life satisfaction, because if you hate your job, it will affect the rest of your life – yes, even your sleep. It affects your relationships, parenting, interactions with outsiders, and energy levels. Being unhappy in your job has many negative consequences on the quality of your life, including shortening your life expectancy.

Some companies tout what's now called work-life balance. Work-life balance is one of those things that works on paper but isn't necessarily attainable in real life. Writers often create entire articles around maintaining a work-life balance, but they aren't system administrators, are they? I just read an article about work-life balance where the writer lists, "Find a job that you're so passionate about that you'd do it for free." I'll make a note of that and then crumble it up and toss it in the trash can. Most of us must work and don't have the option of choosing passion for what we do. I'd rather be a comedian or make films all day. I'd rather write or paint. Maybe if I'd started on something I felt more passionate about earlier in life, I wouldn't have to consider work-life balance.

My former co-workers and I agree that we're too good at what we do to stop now to follow our unrealized dreams. We have too much invested in what we've done. We've accomplished too much. We've created dependencies and expectations that supersede our lofty job satisfaction goals. Sure, we like what we do and were once passionate about it. We assumed that we'd always love what we do. Times have changed. We've changed. We've evolved. And we realize that we are more than just our jobs. We are more than that one-third. Our priorities have shifted from wanting more out of a career to needing more out of life.

I stated earlier that job satisfaction leads to life satisfaction, and perhaps you agreed with that statement, but what if it was the other way around? Life satisfaction leads to job satisfaction. Our lives become so centered around work and career that we focus on it even when not at work. Instead, we should create a happy life outside work and let work take care of itself. In other words, shift focus from "living to work" to "working to live." Once you make that mental shift, your whole perspective changes.

One of my co-workers used to live in Argentina. "They had it right," he said. "They work to live, and we live to work. And they are so much happier." It's a good lesson to learn. Sure, work is important. Work is necessary. Jobs are essential to survival, but it stops there. A career is not your family, it's not your spouse, and it's not you. You've learned to work, and now it's time to learn to live the life you want. The whole point is not to find the perfect job or embark on a quest for job satisfaction. The goal is to build a happy life and to find a job that works for you.

Ken Hess • ADMIN Senior Editor





ADMIN

Network & Security

Features

- 10 Archiving in Microsoft Teams**
Archive teams and channels no longer needed with various on-board tools.
- 14 Cyn.in Groupware**
Set up an environment, exchange knowledge, collaborate on projects, and manage processes with this open source groupware.
- 18 Roadmap for Exchange Server**
The next generation of Exchange 2019 was announced for the second half of 2021, but the release plan was revoked in 2022, and the next Exchange was postponed until 2025. We take an in-depth look at the current timetable.
- 20 Matrix**
This open standard implements secure, decentralized communication and comes with bridges to popular chat apps.
- 26 Sandstorm**
Self-host web-based productivity apps, apply individual permissions, and isolate documents for security, without a hit on productivity.

Tools

- 30 DRBD and Corosync/Pacemaker**
Eliminate single points of failure and service downtime with the DRBD distributed replicated storage system and the Corosync and Pacemaker service.
- 37 MinIO**
This open source, high-performance object storage solution is compatible with the Amazon S3 API and supports multitenant environments.
- 44 Windows Secured-Core Server**
Reduce the attack surface of your system with minimal overhead.

Security

- 58 Phishing Tests**
The Gophish phishing framework lets you set up your own phishing campaigns to identify vulnerabilities and make users aware of these dangers.
- 64 KeePass**
Usernames and passwords play an important role in security. We show you how to set up this password manager and keep it synchronized across multiple devices.

Containers and Virtualization

- 50 AWS Well-Architected Framework**
Develop resilient and efficient cloud infrastructures for enterprise applications.
- 54 Azure Sphere**
Link three vital elements of the Internet of Things - microcontrollers, software, and cloud service - with security built-in.

Service

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



@adminmagazine



@adminmag



ADMIN magazine



@adminmagazine

10 | Teamwork

Collaboration and Communication

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.

Highlights

18 The Future of Exchange Server

Updates through the end of 2023 tend to focus on security, and you can look forward to the new functions that Exchange vNext 2025 will introduce.

37 MinIO

The powerful open source object storage solution stores and manages large amounts of unstructured data; its durability and redundancy features make it an excellent choice for backup and disaster recovery solutions.

76 Zabbix

The strengths of this open source monitoring tool include multiple data collection methods, easy-to-define but extremely flexible problem detection and alerting, and attractive visualizations.

Management

68 Native Prometheus Histograms

Histograms are a proven means of displaying latencies in Prometheus, but until now, they have had various restrictions. Native histograms now provide a remedy.

72 OpenProject

Sensible, comprehensive project management for SMEs with few financial inputs.

76 Zabbix

Comprehensive, highly configurable, but easy-to-use system monitoring.

Nuts and Bolts

86 OpenCanary

The canary in a coal mine has made its way metaphorically into IT security with this honeypot for detecting attacks.

88 Sensor Query Tools

Discover the sensors that already exist on your systems, learn how to query their information, and add them to your metrics dashboard.

94 Performance Tuning Dojo

Standard loads are essential to benchmarking.

On the DVD

Ubuntu 23.04

"Lunar Lobster" Server Edition

Ubuntu Server 23.04 includes nine months of security and maintenance updates until January 2024. This latest version has new hardware support and various performance and security improvements. Upgrades include:

- Linux kernel 6.2
- Updated ca-certificates package
- Toolchain upgrades
- AppArmor updates
- Updated container runtimes



News for Admins

Tech News

NIST Updates Cybersecurity Framework

Major updates to NIST's Cybersecurity Framework (CSF) are underway, with the new CSF 2.0 expected in 2024.

"Based on stakeholder feedback, in order to reflect the ever-evolving cybersecurity landscape and to help organizations more easily and effectively manage cybersecurity risk, NIST is planning a new, more significant update to the framework: CSF 2.0," NIST says. The framework was initially produced in 2014 and updated to CSF 1.1 in 2018.

NIST plans to seek stakeholder feedback throughout the process, and a discussion draft is now available for review (<https://www.nist.gov/system/files/documents/2023/04/24/NIST%20Cybersecurity%20Framework%202.0%20Core%20Discussion%20Draft%204-2023%20final.pdf>). "The modifications from CSF 1.1 are intended to increase clarity, ensure a consistent level of abstraction, address changes in technologies and risks, and improve alignment with national and international cybersecurity standards and practices," according to the draft document.

Check out NIST's Journey to CSF 2.0 website (<https://www.nist.gov/cyberframework/updating-nist-cybersecurity-framework-journey-csf-20>) for the proposed timeline and other information.

Poor Cloud Security Practices Put Organizations at Risk

Many organizations are failing to implement basic cloud security practices and address known vulnerabilities in a timely fashion, according to a new report from Palo Alto Networks' Unit 42.

The Unit 42 *Cloud Threat Report*, Volume 7 (<https://unit42.paloaltonetworks.com/cloud-threat-report-expanding-attack-surface/>), details issues observed in thousands of multi-cloud environments, noting that these "gaps in security are getting more attention from threat actors."

Findings from the report include:

- Security teams take approximately six days on average to resolve a security alert.
- Sixty percent of organizations take more than four days to resolve security issues.
- Eighty percent of alerts in most cloud environments are triggered by just five percent of security rules.
- Seventy-six percent of organizations don't enforce multi-factor authentication (MFA) for console users, while 58 percent don't enforce MFA for root/admin users.
- Sensitive data was found in more than half of publicly exposed storage buckets.

"For threat actors, each workload in the cloud presents an opportunity, and without proper management, organizations are exposed to risk in countless ways," the report says.

ORNL and NOAA Launch New Supercomputer for Climate Research

Oak Ridge National Laboratory (ORNL) has partnered with the National Oceanic and Atmospheric Administration (NOAA) to launch a new supercomputer dedicated to climate science research, which will be one of three NOAA computers operating at ORNL.



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

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

Linux Magazine Subscription

Print and digital options
12 issues per year



► SUBSCRIBE
sparkhaus-shop.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



The new system, called C5, is an HPE Cray machine with more than 10 petaflops (or 10 million billion calculations per second) of peak theoretical performance, which is almost double the power of the two previous systems combined, says the announcement (<https://cleantechnica.com/2023/04/12/new-supercomputer-for-climate-science/>).

The goal of the partnership is to increase NOAA's climate modeling capabilities to advance critical research.

DOE Envisions New High Performance Data Facility

The US Department of Energy (DOE) has proposed and begun the bidding process (<https://science.osti.gov/-/media/grants/pdf/lab-announcements/2023/LAB-23-3020.pdf>) for a new High Performance Data Facility, "which will be a centralized hub of shared datasets and other resources located in one of its HPC labs that connects out through spokes of network to storage and computation in the other centers," says Timothy Prickett Morgan (<https://www.nextplatform.com/2023/03/14/doe-wants-a-hub-and-spoke-system-of-hpc-systems/>).

"The DOE envisions the hub and spoke model will have the HPDF as its foundation, where shared datasets will be centralized and shared across spoke facilities. What the DOE is looking for is more of a system of systems approach than a centralized compute and storage facility, something that looks very much like XSEDE to our eye," Morgan says.

In other news, the HPC (<https://www.admin-magazine.com/HPC/>) computing market is anticipated to grow at a compound annual growth rate (CAGR) of 6.3 percent by 2029, according to a new report from Orion Market Research (<https://www.openpr.com/news/2997234/high-performance-computing-hpc-market-to-witness>).

This "astonishing" rate is driven by factors including a "growing requirement for HPC solutions in various applications, continued diversification, expansion of the IT industry, advances in virtualization, and rising preference for hybrid HPC solutions" the report says.

VMware Updates Tanzu with New Security Features

VMware has announced new capabilities across the VMware Tanzu platform to help customers better manage security, cost, and performance.

According to the announcement (<https://news.vmware.com/releases/vmware-enhances-tanzu-and-aria-platforms>), Tanzu Application Platform 1.5 "enhances end-to-end app security and streamlines developer and platform engineering experiences." New features include "auto-configuration of Transport Layer Security (TLS) and support for external security tools for secrets management."

VMware also announced updates to Tanzu for Kubernetes Operations (<https://tanzu.vmware.com/kubernetes-operations>) that are aimed at helping teams "adopt a 'shift-left' approach to security and greater flexibility" in navigating Kubernetes security policy management.

Microsoft Launches AI-Powered Security Copilot

Microsoft has launched Microsoft Security Copilot (<https://news.microsoft.com/ai-security-2023/>), which aims to "augment the work of security professionals through an easy-to-use AI assistant."

According to the announcement (<https://news.microsoft.com/2023/03/28/with-security-copilot-microsoft-brings-the-power-of-ai-to-cyberdefense/>), Security Copilot "is designed to work seamlessly with security teams, empowering defenders to see what is happening in their environment, learn from existing intelligence, correlate threat activity, and make more informed, efficient decisions at machine speed."

For example, the tool can assist teams "by summarizing data on attacks, prioritizing incidents, and recommending the best course of action to swiftly remediate diverse threats."

Microsoft Security Copilot is currently available through private preview. More information can be found on the website (<https://news.microsoft.com/ai-security-2023/>).

IBM Deploys First Quantum Computer Dedicated to Healthcare Research

IBM has announced deployment of an onsite, IBM-managed quantum computer in the United States. “The IBM Quantum System One installed at Cleveland Clinic will be the first quantum computer in the world to be uniquely dedicated to healthcare research with an aim to help Cleveland Clinic accelerate biomedical discoveries,” according to the announcement (<https://newsroom.ibm.com/2023-03-20-Cleveland-Clinic-and-IBM-Unveil-First-Quantum-Computer-Dedicated-to-Healthcare-Research>).

The Cleveland Clinic-IBM Discovery Accelerator (<https://my.clevelandclinic.org/research/computational-life-sciences/discovery-accelerator>), which is part of a 10-year partnership between the companies, has already generated projects leveraging quantum computing, AI, and hybrid clouds that are aimed at advancing the pace of discovery in healthcare and life sciences, says the announcement.

“Quantum and other advanced computing technologies will help researchers tackle historic scientific bottlenecks and potentially find new treatments for patients with diseases like cancer, Alzheimer’s, and diabetes,” says Tom Mihaljevic, CEO and President of Cleveland Clinic.

LPI Announces IT Security Essentials Certification

The Linux Professional Institute (LPI) has announced the new LPI Security Essentials certification (<https://www.lpi.org/our-certifications/security-essentials-overview>), covering preliminary knowledge in all important fields of IT security.

This certification is intended for those who have completed a first course in IT security, for professionals who want to improve their security skills, and “for anyone who wants to attain a basic competence in the secure use of email, websites, social media, and the devices they use every day,” the announcement says.

To obtain the LPI Security Essentials certificate, candidates must, for example:

- Have a basic understanding of common security threats of using computers, networks, connected devices, and IT services on premises and in the cloud.
- Understand common ways to prevent and mitigate attacks against their personal devices and data.
- Be able to use encryption to secure data transferred through a network and stored on storage devices and in the cloud.

“The exam objectives (<https://www.lpi.org/our-certifications/exam-020-objectives>) cover a comprehensive range of topics, including typical IT security fields like encryption and data security, but also rather uncommon topics like aspects of privacy and the secure use of social media,” says Fabian Thorns, Director of Product Development at LPI.



Archiving teams and channels

Stockpiled

Teams and channels in Microsoft Teams that are no longer needed should be removed promptly; however, to preserve them, they can be archived with various on-board tools. By Thomas Joos

Thanks to Microsoft Teams on-board tools, the collaboration environment can be used to exchange documents. In many cases, the documents in teams and channels will need to be archived in a legally secure and audit-proof manner, at least temporarily. It might also be necessary to reuse documents in a deleted team later or protect them against editing while hiding the team from active view. Before you start archiving data in Teams, you need to think about the differences between backing up and archiving. In general, a backup is not an archive, because the data is continuously overwritten, and virtually nobody keeps a backup permanently. A backup is mostly used in the short and medium term and has the task of restoring deleted or destroyed files. Archiving means long-term storage. Archiving is intended to prevent any changes to data. In this sense, an archive is a collection of data that companies need to store in a tamper-proof manner for legal reasons. Third-party products for backing up Microsoft Teams provide legally compliant storage, but integrated archiving can at least be used in parallel.

In this article, I look at the archiving possibilities that Teams offers out of the box. One massive advantage of using the integrated functions is that they do not delete any data; instead, the team is preserved in its entirety,

although no longer listed as one of the active teams in the Teams client. This operation improves the overview while giving you audit-proof data storage. An added bonus is write protection for chats and all documents in the team.

Integration with SharePoint Online

Teams, OneDrive for Business, and SharePoint are a useful combination of tools in Microsoft 365. SharePoint forms the underpinnings for Teams and OneDrive, with the documents in Teams being stored in SharePoint databases (Figure 1). To do this, Microsoft 365 creates a team website, including a document library in SharePoint Online. In turn, channels in teams get a folder within a team's document library, which plays an essential role in archiving channels and the documents they contain. Of course, deleting a team deletes the associated data from SharePoint Online – but this does not apply to channels. After deletion, the information still remains as a folder in the respective SharePoint library. OneDrive enables data synchronization with terminal devices on the clients. Teams and SharePoint are complementary, not competing, technologies in Microsoft 365. Technologies from SharePoint and Microsoft Office are used. Teams

extends enterprise group collaboration with technology from SharePoint, which is why collaboration on documents is possible in Teams. Typically, the documents you save to the Files tab of a channel are stored in the team's document library in SharePoint Online. However, there is no requirement to store data in Teams with Microsoft. You can also connect external data storage in the cloud, which facilitates archiving because the data in this case is available to multiple teams as long as the appropriate cloud storage is connected. In the Files menu, you can see the connected storage in Teams, such as storage in SharePoint Online. You can select *Add cloud storage* to connect external providers (e.g., Google Drive, Dropbox, Box).

Teams vs. Channels

Generally, you should make sure no objects remain in teams that are no longer necessary to reduce administrative effort, increase security, improve the overview, and streamline backups. Nevertheless, users might need access to certain documents later. Archiving is the proper way to secure the data of groups that are no longer needed against loss, because the data still exists even though the parent team no longer appears in the list of active groups.

Photo by Adis Corovic on Unsplash

One problem with storing documents securely in Teams is that some organizations don't back up all of their data to Microsoft 365 and often don't back up any data at all to Teams. Additionally, team owners are not necessarily administrators but are users with traditional rights. When a team is deleted by the owner, the data backup is then missing – and often the archiving. This unfortunate combination of circumstances can quickly cause the loss of important data.

The cloud platform's recycle bin is not suitable as an archive replacement. Channels moved there will be permanently deleted after 90 days, and deleted teams will be removed by the system after 30 days. In general, you need to distinguish between teams and channels when archiving. In the Teams admin center and with external applications, it is possible to archive groups relatively easily. However, this is not possible with individual channels. Here you need to archive the entire team, including all other channels.

Teams Admin Center

Generally, the Teams admin center is the hub for settings that apply to all teams in the environment. Here,

you also control archiving centrally. The Teams admin center is available in the web browser at the address admin.teams.microsoft.com. To archive complete teams, select *Teams | Manage teams* and click on the corresponding team. Start the process with the *Archive* button. Before execution, you will receive a warning that members will no longer be able to work in the group and all activities will be frozen.

However, it is still possible to add or remove members from the team or update roles, because it should still be possible to control who can read the individual files in the archived team. Archiving only ensures that no changes are made in the chats and documents. In parallel, you can specify that the team's SharePoint page is set as read-only by activating the corresponding option. After you have made the changes, the team and its SharePoint page will be given the status *Archived*.

In the Teams client, archived groups are visible under *Hidden teams*, which allows users to continue to access the team's data, but not modify it. It is possible to hide teams even without archiving. These then appear together with the archived groups under *Hidden teams*. Just hiding a team has nothing to do with

archiving and does not prevent the team from being edited. You control hiding and displaying teams with the team properties from the menu with the three dots in the upper right corner, which is the same way you hide channels in teams.

When archiving, all team data remains in the original location and is only marked as read-only. This state enables a very fast reactivation and at the same time ensures that the data is preserved within the framework of legal regulations. Proceed analogously the other way around: Select the team and choose *Restore*.

Archiving Directly in the Teams Client

Archiving is also available in the group properties of the Teams client – if the user is authorized to do this; some Microsoft 365 subscriptions do not include this option. If the menu item for archiving does not appear in the team's context menu, click the settings gearwheel at the bottom of the Teams client. You can now see all the teams that you can manage as an owner or admin. The menu with three dots for each team normally has an *Archive team* option (Figure 2). Again, you need to set the SharePoint site to read-only.

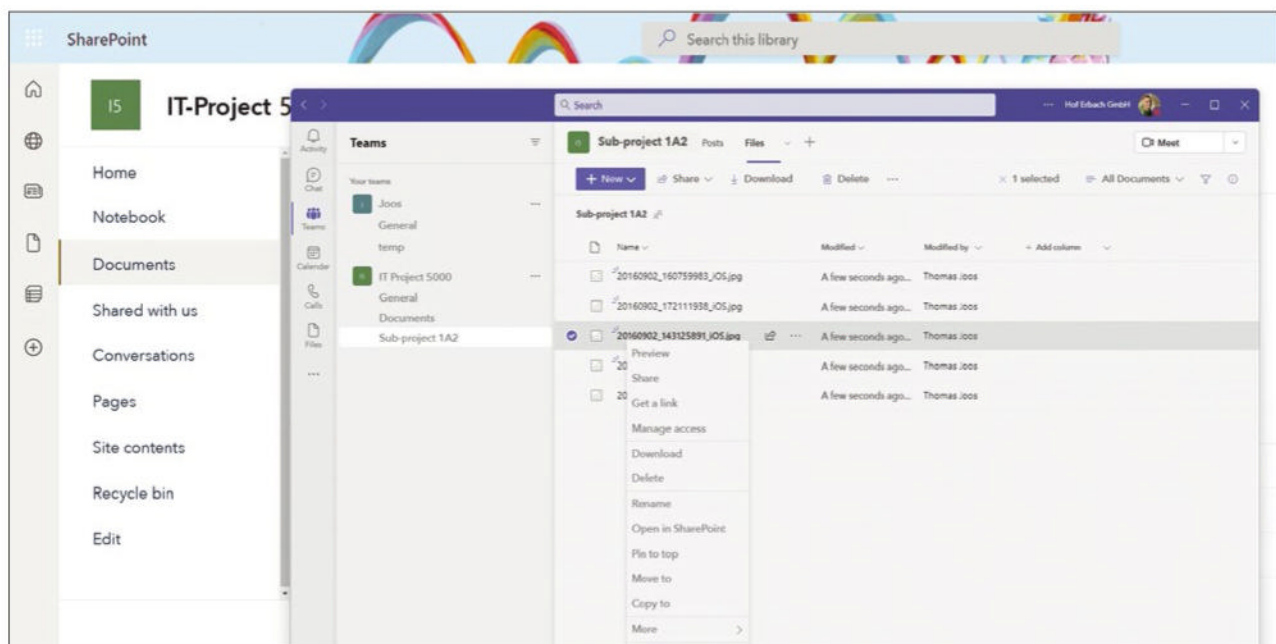


Figure 1: Documents that users save to teams and channels eventually end up in the team's document library in SharePoint Online.

Provided you have the right subscription and matching authorizations, archived teams can also be reactivated in the Teams client. The two options for this are also available in the menu with three dots. The archived groups in the Teams client are listed in *Archived*. To reactivate, just select *Restore team*. After restoring the team, the members can continue to edit the documents.

Archived teams can be searched and the documents read. However, changes are ruled out. Teams locks down all team activities after archiving. Users are no longer allowed to start new conversations or reply to posts on a channel, add or remove channels, edit settings, or add apps. However, as a team owner, you can still add or remove members, update roles, and delete, renew, or restore an archived team.

Restoring Teams and Channels

Team owners and admins can delete teams. In this case, the team's data will remain available in the Microsoft 365 recycle bin for 30 days. However,

as I mentioned earlier, this is not long-term archiving. You can use the menu in the Teams client to delete or press the *Delete* button after selecting the team under *Teams | Manage teams* in the admin center menu. Deleting wipes out the group mailbox, calendar, and all the files stored on SharePoint Online. All data in OneNote, Planner, Power BI, or Stream are lost during this process. Individual channels can also be deleted if a user has the right authorization. However, team owners can recover deleted channels with relative ease by clicking *Manage team* in the team's context menu and then clicking on the *Channels* tab. You will then see the deleted channels under *Deleted*. At this point, you can also see when the channel was deleted, and you can reactivate the channel with the *Restore* menu item in the recycle bin. During this process, Teams restores the files that were stored in the channel. Strictly speaking, the files were never deleted, but only hidden in Teams.

When you delete a channel, SharePoint keeps the channel's files as folders in the team's document library.

The users are just no longer able to access the documents. To ensure that these files are no longer present, you need to delete the folder in SharePoint itself, but if you then try to restore the channel, the documents are lost because the process in Teams can no longer access the SharePoint source.

When restoring a channel, Teams discovers that the folder for the channel is missing from the Teams document library and creates a new one. Nothing is recovered from the SharePoint recycle bin. However, you can manually restore the deleted channel folder from the recycle bin in SharePoint. Before you do so, you need to delete the folder that was automatically created before you restore it with the data from the SharePoint recycle bin.

Administration with PowerShell

When deleting teams, Microsoft 365 moves the data to the recycle bin to enable their recovery with the *Teams and groups | Deleted groups* item in the Microsoft 365 admin center.

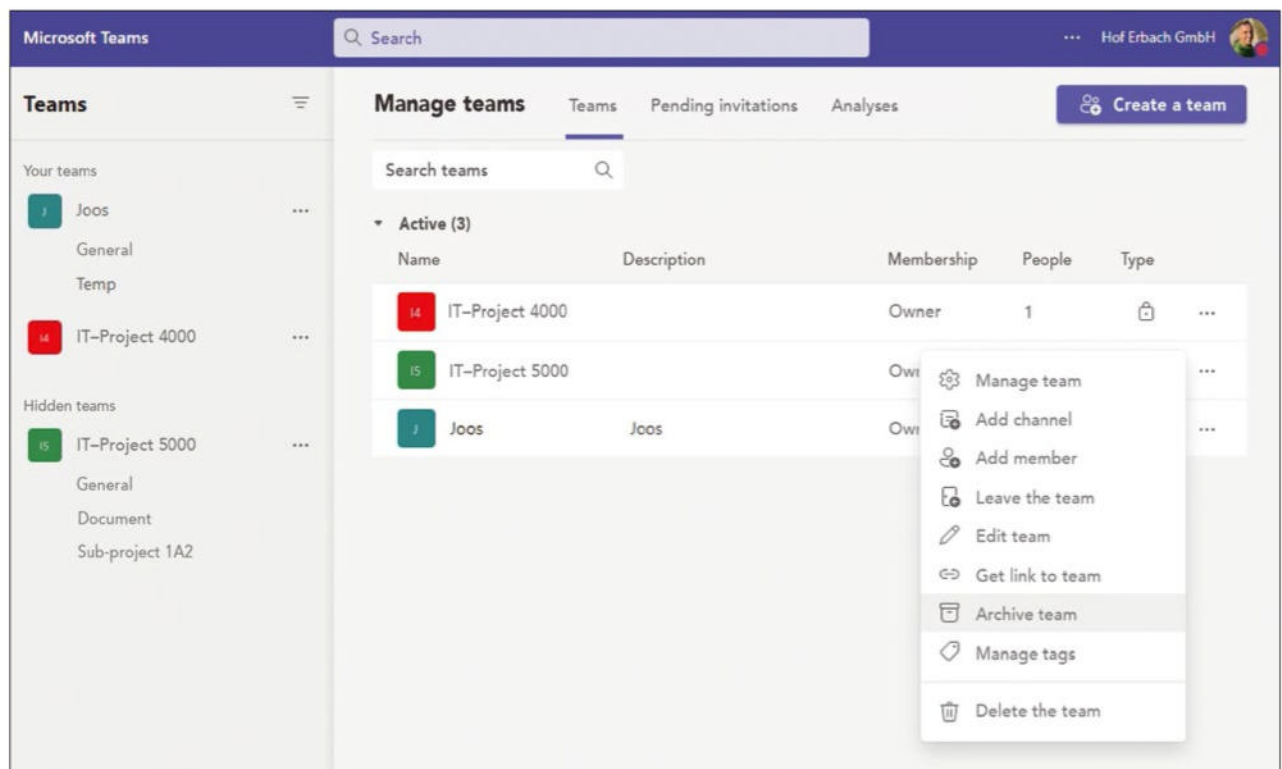


Figure 2: You can archive clients in the Teams contextual menu.

Teams are constructs that build on Microsoft 365 groups. When you delete a team, you delete the associated Microsoft 365 group.

This arrangement explains why recovery takes place in the Microsoft 365 admin center, not in the Teams admin center. You will find the deleted groups and the connected teams there. *Restore group* starts the process.

PowerShell is also a path for recovery. The process relies on the AzureADPreview module. To ensure that you are using the latest version, first uninstall the two associated modules then install the latest version:

```
Uninstall-Module AzureADPreview
Uninstall-Module AzureAD
Install-Module AzureADPreview
```

When done, connect to Azure AD and query the deleted teams and groups from Microsoft 365:

```
Connect-AzureAD
Get-AzureADMSDeletedGroup
```

The `Get-AzureADGroup` cmdlet lets you view all the groups and Teams in PowerShell. The recovery is based on the team ID, which you can see in PowerShell; for example,

```
Restore-AzureADMSDeletedDirectoryObject -id 9c5d6112-f57d-4b7e-8ecc-548e9f79cf7d
```

Note that it may take several hours for a recovered team to work correctly and for all data to be restored.

You can also use PowerShell to archive teams. To do this, you need the PowerShell module for Teams, which you can install with the

```
Install-Module -Name MicrosoftTeams -Force -AllowClobber
```

command. Next, log in by typing

```
Connect-MicrosoftTeams
```

In the active PowerShell session, import the module and display all the teams first (Figure 3):

```
Import-Module MicrosoftTeams Get-Team
```

You can see the team IDs here. You need the ID to archive a team:

```
Set-TeamArchivedState -GroupId 5a8edffb-43ec-4dbc-bf12-4b50d2c65240 -Archived:$true
```

You can check the value of `Archived` again if you query it with `Get-Team`. To make sure that the associated SharePoint page is read-only, type

```
Set-TeamArchivedState -GroupId 5a8edffb-43ec-4dbc-bf12-4b50d2c65240 -Archived:$true -SetSpsiteReadOnlyForMembers:$true
```

```
-Archived:$true -SetSpsiteReadOnlyForMembers:$true
```

To stop archiving again, use

```
Set-TeamArchivedState -GroupId 5a8edffb-43ec-4dbc-bf12-4b50d2c65240 -Archived:$false
```

to set the value of `-Archived` to `$false`.

Conclusions

Archiving teams is more important than many Microsoft 365 users might think. It is not enough to create a plain backup of a team, nor can you simply delete teams for archiving purposes and rely on storing them temporarily in the recycle bin. Teams can be archived just as easily with the Teams admin center or the Teams client, along with support for PowerShell.

Whatever the case, the process owners in the enterprise need to be up to speed on how backing up and archiving works in Microsoft Teams. ■

The Author

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

```
PS C:\Users\thoma> Import-Module MicrosoftTeams
PS C:\Users\thoma> Update-Module MicrosoftTeams
PS C:\Users\thoma> Get-Team
```

GroupId	DisplayName	Visibility	Archived	MailNickName	Description
c8a81d2a-25d0-4417-a991-ef16eda0ca4c	Joos	Private	False	Joos	Joos
5a8edffb-43ec-4dbc-bf12-4b50d2c65240	IT-Project 5000	Private	False	IT-Project5000	

```
PS C:\Users\thoma> Set-TeamArchivedState -GroupId 5a8edffb-43ec-4dbc-bf12-4b50d2c65240 -Archived:$true -SetSpsiteReadOnlyForMembers:$true
```

GroupId	DisplayName	Visibility	Archived	MailNickName	Description
5a8edffb-43ec-4dbc-bf12-4b50d2c65240	IT-Project 5000	Private	True	IT-Project5000	

```
PS C:\Users\thoma> Get-Team
```

GroupId	DisplayName	Visibility	Archived	MailNickName	Description
5a8edffb-43ec-4dbc-bf12-4b50d2c65240	IT-Project 5000	Private	True	IT-Project5000	
c8a81d2a-25d0-4417-a991-ef16eda0ca4c	Joos	Private	False	Joos	Joos

Figure 3: You can also archive individual teams in PowerShell and disable archiving again.



Cooperation with Cyn.in

Fast Action

Cyn.in open source groupware focuses on connecting employees quickly and easily. We show you how to set up a Cyn.in environment, exchange knowledge, collaborate on projects, and manage processes. By Holger Reibold

Software's primary task is to provide users with tools that enable them to perform typical tasks efficiently, getting more done in less time. To that end, Cyn.in [1] helps you improve communication and optimize the exchange of knowledge.

A Virtual Appliance

The Cyn.in Python-based environment is based on a Debian system, including an Apache web server, if you use the virtual appliance. From a technical perspective, Cyn.in is an add-on for the Plone content management system, which in turn is based on Zope. In the background, the Zope Object Database (ZODB) manages the objects.

The easiest way to evaluate the software is to use the virtual appliance, which is available for download from SourceForge. Cyn.in is not particularly resource hungry. According to the developers, a standard system with 8GB of RAM and 10GB of free disk space is all you need, plus

Internet access, of course. Working with the ISO file, install the associated Debian system along with the required components.

After completing the installation, it's time to open the wizard in the browser for web-based configuration by typing `http:// <IP-address-of-the-Cyn.in-system> :8004`. The secured URL (`https`) is available on port 8003. The configuration wizard shows a login dialog for initial access, which accepts the default username/password `admin/password`. The wizard guides you step by step through the rest of the configuration. The first step is to assign a new password to the administrator. Clicking on *Save* will take you to the next configuration section, which is email notifications. Here, you specify the email server and the sender that will send the notices. You will want to create at least one recipient who will be notified by email about system-critical events.

Another click on *Save* takes you to network configuration, where you define

the hostname of the Cyn.in system and the gateway. Alternatively, you can leave this to the DHCP server on your network. Customization options for DNS and Ethernet then follow. If any changes are made to the network, the virtual appliance requests a reboot. If you accept the predefined settings, *Save* takes you to the proxy configuration. Because Cyn.in is a web-based environment, you can customize the HTTP/S proxy settings. If the proxy server requires authentication, store the credentials in the wizard. After saving the data, Cyn.in welcomes you to the admin console, which shows the status of the environment and lets you carry out typical admin tasks.

Completing the Basic Configuration

Cyn.in distinguishes between system administration of the underlying Debian system and management of the groupware environment. System-related functions are limited to a few customization options, accessing

Photo by Jo Coenen - Studio Dries 2.6 on Unsplash

appliance logs, scheduling system reboots, and applying rollbacks. You can also check for system updates and discover whether they are installed.

Cyn.in administration happens on the front end, which can be accessed from the same hostname or the assigned IP address, but always through the default port. As part of the initial configuration, Cyn.in creates a site administrator (the login is *siteadmin/secret*). Each Cyn.in account has its own profile, for which you want to change the passwords. You can add to the profile in *Personal Preferences*, where you save, for example, the email address and a profile picture. Provided you have administrative rights, you can also deactivate its listing in a search.

Although you configured some email-specific settings during the system setup, they only relate to warnings from the Debian system. To make sure the groupware environment also interacts with the message server, you need to adjust the email settings for the site administrator. To do this, switch to the *Mail Setup* menu in the environment settings; you will recognize these by the wheel symbol in the header. Cyn.in assumes by default that you will be using the local email server for sending. If you specified a specific hostname during system setup, you need to specify a valid email address that the Debian system provides as the *Site From address*.

User management is one of the other central tasks in Cyn.in, and it has been made easy thanks to the tabular overview and the integrated search. Return to the settings under *User Management* and create your first user by following the *Add New User* link. Cyn.in requires a real name, the username, and an email address. Click *Register* to create the account and generate a registration email, which the user has to confirm. The site administrator can assign users to groups (e.g., Editor, Reader, Manager) in user management. Group management is organized in the *Group Management* menu.

Collaboration Modules

For communication and collaboration, the environment provides various modules that the site administrator can enable or disable as needed. Users have easy access to basic features, such as the shared calendar and discussion forums, through the front end. For example, you can generate task- and team-specific file repositories in which to store documents. The associated functions are available under *Files*. The system automatically generates a version history. Access rights are controlled with a granular rights system. Galleries for images and audio and video files are also available to support collaborative work on projects (Figure 1).

Cyn.in's various functional areas are accessible in configurable views. The blog module plays a central role; the developers believe it can develop into a genuine think tank with a value-generating potential. The associated functions are available under *Blog*. To create a new entry, click on the *New* icon. The idea is to not just use the blog function to gather ideas, but to develop ideas actively with comments and other activities.

Because many groupware systems do not do justice to this knowledge management aspect, the developers integrated a *Wiki* module that lets

you create a cross-departmental or cross-company knowledge database. In the familiar style of Wikipedia and other systems, arbitrary content can be linked in this way. Content development is simplified by the integrated WYSIWYG editor that autosaves changes every minute by default.

Who Works Where

Workspaces are a defining element of the groupware environment, and you can generate these from the menubar. For example, you can assign separate areas to marketing, human resources, or sales and assign the employees to these areas from the *Spaces* configuration.

The workspace settings are where you create workspace-specific workflows and specify functions and workflows. Adding content to the workflows at the same time is a good idea. As with all other Cyn.in modules, you assign the desired applications and content types to the various workspaces in the module settings. The sharing function, which is available globally on the system in the menubar, is also of benefit. For example, to share content, you just follow the *Sharing* link and specify the content and the group.

The workspace function provides another special feature, wherein you

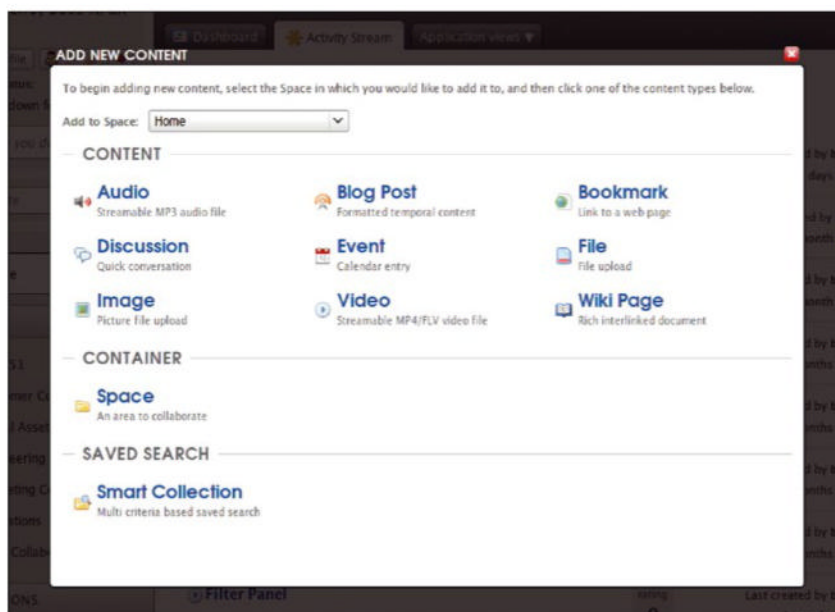


Figure 1: Cyn.in can be extended to include more functions and workspaces, if required.

can create structured sub-spaces in a workspace, and employees can create their own workspaces – including sharing permissions. By doing so, you can reflect the hierarchies that exist in the enterprise. With each new space, Cyn.in automatically adjusts the workspace hierarchy. Environment users navigate the entire structure from *<Cyn.in Site>/Spaces* on their homepage and can interact with co-workers wherever they have appropriate permissions. The apps available in each workspace are also configurable.

Social Applications

Groupware environments basically aim to optimize communication and improve the exchange of data and information. The Cyn.in system also does justice to this requirement by implementing contextual workspaces. You can create them in the *Spaces* configuration, which is available on the homepage. The goal of this function is to generate crowd knowledge. An Ajax-based commenting system is used in the contextual areas, which means that knowledge can be gleaned from exchanges between employees.

From the system-wide user directory, employees can easily contact co-workers, exchange information, and collaborate on content in workspaces. Typical status information shows who is available and when. Cyn.in also uses “infocards,” which contain general contact and business information and other relevant information. This area also dynamically consolidates user activities, content, and posts generated by users on a single

dashboard. Posts and other content elements can be rated by users, and the highest ranking elements are highlighted in a word cloud.

Individual Pages for Users

To increase employee productivity, Cyn.in promises several supporting features, such as individual employee dashboards for personal employee workspaces through various portlets that implement, say, content aggregators, navigation systems, statistics, and graphics. Access is from the *Spaces* entry on the homepage and the matching *Space* setting. You can also benefit from the compatibility of Cyn.in dashboards with Plone portlets, which you can integrate into the environment and deploy globally on the system.

Activity streams let you implement cross-environment messages about changed content elements. When you do so, you decide which users are automatically informed about specific changes. The supported parameters include users, content types, tags, and dates. Basically, this classic feed system comes with a task-specific distribution capability. Besides this info system, you can also configure classic email notifications in the Cyn.in system settings.

The possibilities for client connection in Cyn.in are quite limited. Besides standard web access, the groupware system only supports WebDAV, which allows desktop clients direct access to storage areas. To configure the settings, you go to the *User Management* area. The creator, Cynapse, had been working on a desktop client, but the work has been discontinued.

Organized Collaboration

Cyn.in offers several functions to improve organizational tasks. The role-based security system first ensures that only authorized employees can use certain functions and create and edit content elements. The settings are available under *Rules* in the menubar. Their use is self-explanatory: You assign dedicated access rights for different actions and elements to users and groups. To share content, just follow the *Sharing* link.

Cyn.in supports enterprise workflows with an integrated workflow engine that executes *n*-level workflows that can be applied to any space or content type. Workflows define a formal process in groupware that targets specific collaborative activities within environments. By default, Cyn.in comes with a few simple approval workflows that you access under *Actions*. In principle, you can also extend the environment to include complex business-specific workflows with Python programming.

Conclusions

Cyn.in offers useful features for standard tasks, but it is difficult to avoid noticing that the system has reached a certain age. Alternatives like EGroupware offer more functionality and flexibility, and connecting clients in Cyn.in could be easier. In principle, however, nothing stands in the way of collaboration in the enterprise with this free software. ■

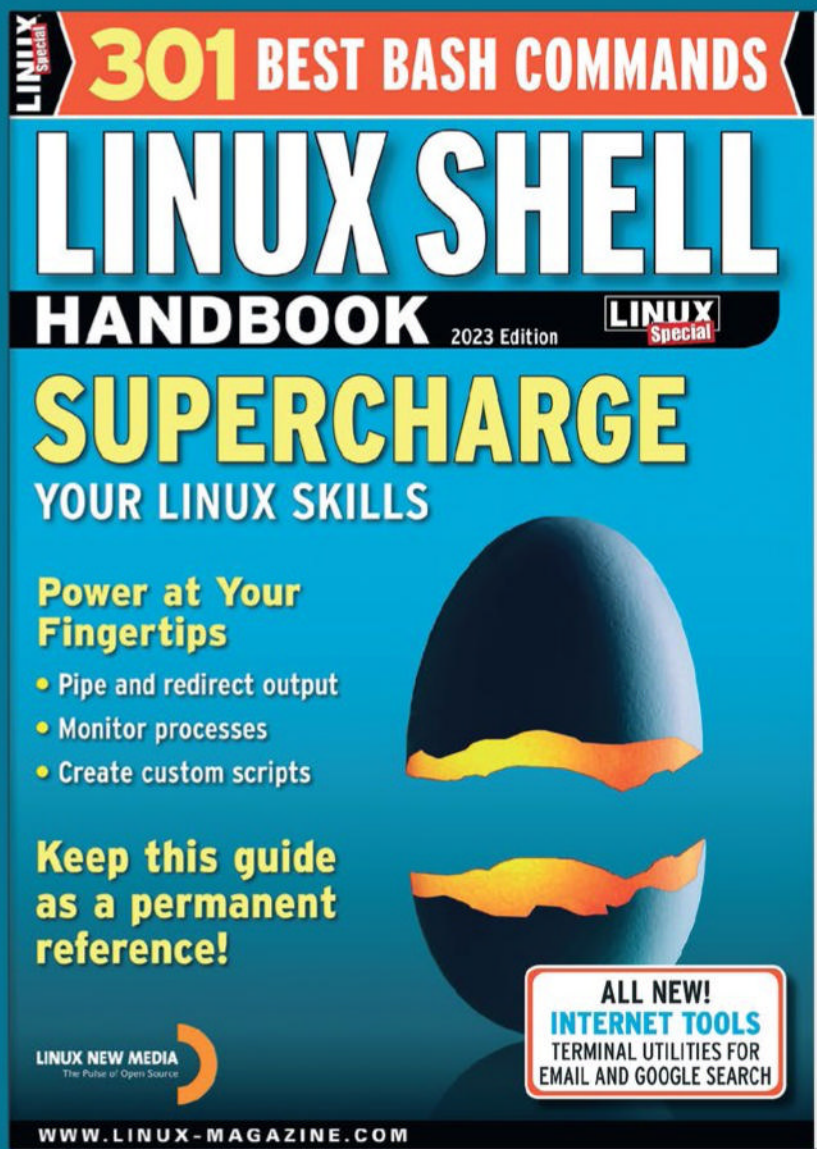
Info

[1] Cyn.in:
[\[https://sourceforge.net/projects/cynin/\]](https://sourceforge.net/projects/cynin/)

THINK LIKE THE EXPERTS

Linux Shell Handbook 2023 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:
sparkhaus-shop.com/specials



Exchange Server through 2025

Holding Back

The next generation of Exchange 2019 was announced for the second half of 2021, but the release plan was revoked in 2022, and the next Exchange was postponed until 2025. We take an in-depth look at the current timetable. By Christian Schulenburg

Microsoft has broken the triennial update cycle for Exchange and does not expect to release a new version until the second half of 2025. By then, seven years will have passed since the release of Exchange 2019 in 2018. According to the people in Redmond, this postponement is mainly the result of the large number of security problems. For example, the Hafnium exploit in early 2021 exposed a vulnerability that affected many on-premises Exchange servers and highlighted how important it still is to maintain your on-premises Exchange installation. It took some time to eliminate the vulnerability on the majority of servers.

To close security gaps more quickly in the future, at least temporarily, Microsoft decided to integrate the Exchange Emergency Mitigation Service (EEMS) directly with Exchange. Moreover, the Antimalware Scan Interface (AMSI), already included in Windows 2016 and 2019, was added to Exchange 2016 and 2019, allowing AMSI-enabled antivirus software to scan the content of HTTP requests sent to Exchange servers and block malicious requests before they are processed.

Additionally, Microsoft has repeatedly released security updates (SUs), which led to a change in 2022 in the way SUs are installed [1]. All told, substantial resources have gone into stabilizing Exchange, ultimately at the expense of developing the next version.

Route to the New Exchange

Companies still using Exchange 2013 that had hoped to switch to the new version right away will now have to switch to Exchange 2019 very quickly because support for Exchange 2013 ended in April 2023. Exchange 2016 left mainstream support and moved to extended support in October 2021, meaning that Exchange 2016 also should not be used any longer. During this phase, Exchange will not see cumulative updates, but security updates will continue to be released. Exchange 2019 will move to extended support as early as January 2024. For both versions (i.e., 2016 and 2019), this support will then expire at the same time in October 2025. Therefore, Exchange 2019 has a shorter release time of just seven years compared with 10 years for other versions. The release of the

next Exchange Server, vNext, in the second half of 2025 does not allow for much leeway when upgrading (Figure 1). The time window to keep support in place is very short. However, the announcement that an in-place upgrade will again be possible with Exchange 2019 is good news. The update should be possible without the need for new hardware or to move mailboxes. Microsoft plans to announce more specific details on requirements, features, and prices in early 2024. Until the new Exchange is released, the recommendation from Microsoft is that all customers migrate to Exchange 2019. Redmond will release feature updates for 2019 in the near future that are no longer available for previous versions. In the announcement, Microsoft not only refers to the next version, but goes one step further, because with the new release, Exchange is moving to the Modern Lifecycle Policy, with no end-of-life date and support for as long as market demand exists. Development therefore continues after Exchange vNext – a forced switch to Exchange Online because of a lack of on-premises versions is off the table.

Photo by Jon Tyson on Unsplash

Feature Updates for Exchange 2019

Microsoft is not talking about the new features for vNext but points to 2024. However, the group goes into far more detail about the functions it wants to implement in Exchange 2019 before the new version arrives.

One important point is modern authentication (MA), which is now fully implemented in Exchange Online – after basic authentication was disabled in October 2022 – and is the only way to log in. This feature ensures an essentially more secure login and supports the use of multifactor authentication, smart cards, and client certificates. For hybrid environments, Microsoft has also enabled MA. In 2019, however, it was determined that MA would not be supported for on-premises-only installations and that a hybrid environment would be the only option for (partially) on-premises use. Microsoft has now moved away from this statement and work is already underway to implement MA in Exchange-only setups. A more detailed timeline is expected to be released later this year.

When it comes to security, TLS also plays an important role. For example, Windows 2022, as the underpinning of Exchange 2019, natively supports TLS 1.3, but not Exchange itself. Support is now firmly scheduled for 2023 and contributes to greater security in communications.

By now, every Exchange admin should be aware of the importance of keeping Exchange up to date. That said,

checking the patch level has been difficult in the past, and PowerShell scripts have often been used as an aid. In the future, administrators will be able to see which servers need to be updated directly in the Exchange Admin Center from a Software Updates Dashboard. Exchange Online got the ball rolling at the end of 2022 with an overview of servers in hybrid environments. Availability of this feature is also expected for on-premises Exchange environments in early 2023.

The EEMS service mentioned at the beginning of this article has also been updated. Although administrators now have to undo rules triggered by EEMS manually, it should be possible in the future to delete rules that are no longer needed with a script. The script also is expected in 2023.

The update process itself is being improved to help avoid changing security postures. Whereas changes to the `web.config` or `sharedweb.config` file were overwritten in the past by installing a cumulative update (CU) and needed to be updated again, Exchange will keep the changes in the future. This adjustment was already announced for the end of 2022 or the first half of 2023.

Constantly changing settings (e.g., for the email size limit in Outlook on the web (OWA)) should therefore be a thing of the past.

The Hybrid Configuration Wizard (HCW) also sees a minor function update. Thus far, when you re-run the HCW, it has gone through all the steps and prompted you for settings that were defined during the initial configuration. When you reconfigure, settings

made in the meantime could be lost. In the future, the wizard will let you skip unnecessary steps.

Conclusions

The hook tempting admins to migrate to Exchange 2019 has been cast, with some interesting features in the works for the current version. Updates through the end of 2023 tend to focus on security. Whether there will only be security updates from 2024, meaning that no feature updates will appear for the only available Exchange products over an extended period of time, remains unclear. Admins can also look forward to the new functions that Exchange vNext 2025 will introduce. However, with the possibility of in-place updates and the full pipeline for Exchange 2019, expectations should not be set too high. ■

Info

[1] Exchange security updates: [\[https://techcommunity.microsoft.com/t5/exchange-team-blog/new-exchange-server-security-update-and-hotfix-packaging/ba-p/3301819\]](https://techcommunity.microsoft.com/t5/exchange-team-blog/new-exchange-server-security-update-and-hotfix-packaging/ba-p/3301819)

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.

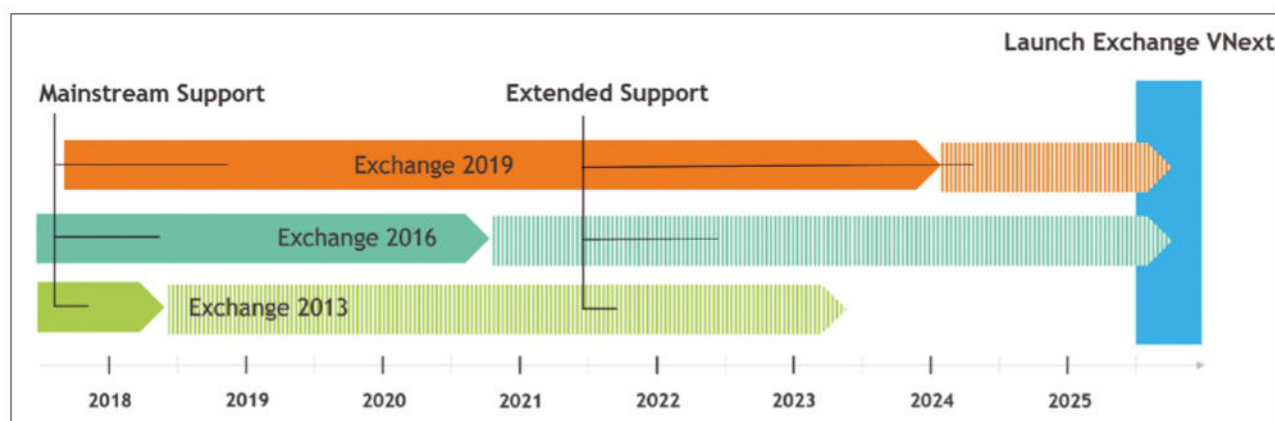


Figure 1: The end of Exchange 2016 and 2019 overlaps with the planned launch of Exchange vNext – with hardly any time for migration.

Run your own chat server

Choosing the Red Pill

The Matrix open standard implements secure, decentralized communication and comes with bridges to popular chat apps.

By Andreas Stolzenberger

Millions of users communicate with central chat services such as WhatsApp or Telegram, entrusting their messages to a centralized platform and its operators. If this meets your needs, you can take the blue pill and pass on to the next article. But if you are interested in Matrix as an alternative, which enables chats without an external provider but still offers the established services, take the red pill and proceed.

The choice between the red and blue pills is not quite as difficult for chat systems as it was for saving humanity in the 1999 movie *The Matrix*. However users are often faced with the decision as to whether they want to trust their chat and group communications to large corporations like Meta (Facebook, WhatsApp) and Alphabet (Google) or a corporate construct like Telegram, or whether they would rather not. If you appreciate the convenience of a modern chat application with groups but prefer to back up your data yourself, Matrix [1] is a serious alternative. However, setting it up requires some work and, of course, a dedicated server.

Open Protocol

Matrix itself is not software. It is an open source, end-to-end encrypted protocol for chat and real-time communication. As an open standard, it ensures that different software implementations like Element (client) and Synapse (server) remain compatible with each other. The principle of Matrix is simple: You run your own Matrix server for an Internet domain (e.g., *domain.com*) and create your users with names following the pattern *@name:domain.com*.

Up to this point, everything looks be very simple and you seem to have yet another closed chat service. However, Matrix lets you publish the server on the Internet (federation) so that users of different Matrix servers can connect to each other. In simple terms, Matrix follows the example of the Simple Mail Transfer Protocol (SMTP) mail server, which distributes local messages on the closed network and forwards external messages to the respective SMTP servers in other domains. As with mail services, self-hosted solutions with their own servers are mixed with cloud

offerings like *matrix.org*. However, Matrix users are spared the spam that mail users have to live with every day because, unlike SMTP, the Matrix protocol does not deliver messages without being asked. If a user of a third-party domain wants to make chat contact, the recipient must agree.

Of course, if you set up your own Matrix server, you then face the problem that many of your existing contacts will still remain on WhatsApp, Facebook Messenger, or Telegram. Alternatively, you could work with closed systems like Discord or Slack. Matrix offers a whole series of bridges for this purpose that connect to the third-party services for the respective user and forward messages to Matrix. With the correct configuration, you then only need to use a single Matrix chat client through which you can communicate with other Matrix users, as well as with all other chat platforms, through the one Matrix server.

In this article, I show you how to set up your own Matrix server with the open source Synapse software, use matching clients, and set up a bridge

Photo by Volodymyr Hryshchenko on unsplash

for WhatsApp. Of course, Synapse can be run on a virtual machine (VM) with Debian or Fedora Linux, but I deploy the server in a Podman container (you can also use Docker), which then works largely independently of the server operating system.

Preparations

Before you can set up the Synapse server for Matrix, which is written in Python, you need to take care of a few preparations. The Matrix service in the container uses the HTTP protocol on port 8008 or HTTPS on port 8448. You could now release port 8448, route the Matrix traffic there, and give the Synapse server a valid Let's Encrypt certificate, for example. However, that is more work than necessary.

In this example, the Synapse service runs on a rented server (Hetzner) with a single IP address. The server is also used by a number of other services. The incoming traffic is therefore distributed by an NGINX reverse proxy to the various service containers according to the name of the services. Packets for *service1.domain.com* go to a different container than requests to *service5.domain.com*. Additionally, the NGINX server handles secure socket layer (SSL) termination for all services and manages the domain's Let's Encrypt certificate. HTTP is then all you need between the proxy and the services.

For this Synapse example, I used the DNS name *matrix.domain.com* and the regular HTTPS port 443. Depending on how you run your Docker or Podman setup, NGINX forwards traffic to an internal bridge IP address or mapped port. In this case, it is a heterogeneous infrastructure that is gradually migrating services from traditional VMs to containers, which is why these containers run on the same bridged network as the VMs and why each container has its own internal IP address. Another reason is security. Access to services on an internal network can be controlled and

monitored more easily through the firewall than if your containers are bound directly to the externally accessible interface by port mapping. The reverse proxy configuration for the Synapse server (*/etc/nginx/conf.d/matrix.conf*) looks something like [Listing 1](#). All HTTPS traffic for *matrix.domain.com:443* now reaches port 8008 of the container over the internal bridge IP 192.168.122.26. If you are running containers without a local bridge, the entry is:

```
proxy_pass http://127.0.0.1: 8008;
```

To the outside world, your Synapse server goes by the name of *matrix.domain.com*, but the service should use names like *@user1.domain.com*, not *@user1.matrix.domain.com*. By extension, you need to run your Synapse server with the *domain.com* configuration and at the same time tell third-party servers that chat messages for *domain.com* should be routed to the server on *https://matrix.domain.com:443*.

For discovery purposes, the Matrix protocol uses two options. A server that wants to contact another domain first tries to locate the target server with a DNS query. Synapse follows the example of SNMP and publishes the mail server of a domain with the MX DNS record. Matrix asks for an SRV entry named *_matrix._tcp* (i.e., in this example, for *_matrix._tcp.domain.com*). The response must then be *matrix.domain.com* and port 443.

However, not all domain owners can easily create or modify DNS records. If the DNS query for the SRV record fails, the Matrix protocol takes a different tack. It runs a REST API request over HTTPS to the domain name with the URL */.well-known/matrix/server* (in this case to *https://domain.com/.well-known/matrix/server*). Matrix expects an HTTP 200 response in JSON format. In this case, this response is also passed by the NGINX reverse proxy. The configuration is usually in the basic configuration of the NGINX server at */etc/nginx/nginx.conf* ([Listing 2](#)).

Configuring Synapse

To configure the Synapse server, you first need a configuration file. The tool kindly creates these when first launched. First create a directory on the container host where Synapse will store its data (i.e., */var/pods/synapse* in this case), and then start the container with:

```
podman run -it --rm --name synapse \
--volume /var/pods/synapse:/data:Z \
-e SYNAPSE_SERVER_NAME=domain.com \
-e SYNAPSE_REPORT_STATS=yes \
docker.io/matrixdotorg/synapse:
latest generate
```

If you are using Docker instead of Podman, you need to run the `docker run` command as root – with the `Z` at the end of the volume specification on systems with SELinux active. The `SYNAPSE_REPORT_STATS` switch lets Synapse send anonymous operating

Listing 1: Synapse Reverse Proxy

```
server {
    listen 443 ssl http2;
    server_name matrix.domain.com;

    ssl_certificate /etc/letsencrypt/live/domain.com/
        fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/domain.com/
        privkey.pem; # managed by Certbot
    [...] (SSL and LOG parameters)
    location ~ ^/_matrix/_synapse/client$ {
        client_max_body_size 100M;
        proxy_pass http://192.168.122.26:8008;
        proxy_set_header X-Forwarded-For $remote_addr;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_set_header Host $host;
    }
}
```

Listing 2: NGINX Reverse Proxy

```
server {
    listen 443 ssl http2;
    server_name domain.com;
    root /usr/share/nginx/html/blank;
    [...] (SSL configuration)
    location /.well-known/matrix/server {
        default_type application/json;
        return 200 '{"m.server": "matrix.domain.com:443"}';
    }
}
```

statistics to the developers. If you do not want this, specify no here. After the command executes, you end up with files `homeserver.yaml` and `log.config` and a signing key in the `/var/pods/synapse` directory. In the YAML file you will find the basic configuration for the service, the database, and a `registration_shared_secret`. Keep this shared secret in a safe place – you will need it to register new users – but remove it from `homeserver.yaml` after the initial setup.

As a database, Synapse uses SQLite in the basic configuration. Although this setup is fine for small test installations, if you set up a Synapse server for dozens of users and chat rooms, you will definitely want to use a PostgreSQL server instead. The Synapse documentation describes in detail how to set up this database, but it does not necessarily have to happen immediately during the first trial run. The documentation also describes how to transfer an existing SQLite database to PostgreSQL retroactively and change the Synapse server to

match. In this test setup, I decided to stick with SQLite.

With the appropriate configuration, you can now launch your Synapse container. Depending on the network configuration, you run it on the host IP with the parameter `-p 8008:8008` to make the HTTP port available on 127.0.0.1:8008. This setup uses Synapse on the bridge with

```
-- net virt_net --ip 192.168.122.26
--mac-address 53:54:C0:A8:7A:1A
```

(i.e., with its own IP address). Then, create a service definition for the Synapse server so that systemd can run the pod at system startup (see the “Synapse.service for Systemd” box).

Creating Users

To register new users, go to the command-line interface (CLI) of the host in the running Synapse container and enter:

```
podman exec
-it synapse register_new_matrix_user
```

```
http://localhost:8008
-c /data/homeserver.yaml <username>
```

Specify a password and give the initial user admin rights. In a larger setup, of course, you will not generate all users manually on the CLI. Synapse has several authentication plugins; you can use single sign-on with different ID providers or add the `enable_registration = true` target in the `homeserver.yaml` file. Be careful, though, because it allows any user with access to the server to create an account. You will only want to enable this function on the local network as long as the server is not yet accessible over the Internet. Matrix separates the client from the server. The Synapse server itself does not have a web user interface (UI). Administration is by the CLI, the REST API, or external tools with access to the admin REST API. You can find a number of different clients for using the service. Element is probably the most popular, and you can use it as a web service in your browser by going to <https://element.io>, or you can download client apps for Windows, Linux, or macOS from the website. Element is also available free of charge for iOS and Android in the Apple and Google app stores. In the Element client, instead of *matrix.org*, enter the URL of your Matrix server (in this example, <https://matrix.domain.com>). After logging in with the initial account and the assigned password, Element prompts you to set up a secure backup and create a secure key for end-to-end encryption. Be sure to store it in a safe place after generating this key. Now you can only connect new clients to your account if you either specify this key or use two-factor authentication on a previously registered client. At this point, it makes sense to link a mobile client on your phone to the account. You can then unlock additional clients by QR code that Element scans and verifies on your phone.

User Interface for Admins

In addition to Element as a messaging client app you can find a few

Synapse.service for Systemd

The service definition is located in `/etc/systemd/system/synapse.service`. If you enable the service with

```
systemctl enable synapse.service
```

the host starts the container automatically at boot time. This service file uses the bridge network `virt_net`. If you work without a bridge, leave out the `--net`, `--ip`, and `--mac-address` lines and add `-p 8008:8008` instead:

```
[Unit]
Description=synapse
After=network-online.target
Wants=network-online.target

[Service]
ExecStartPre=mkdir -p /var/pods/synapse
ExecStartPre=/bin/podman kill synapse
ExecStartPre=/bin/podman rm synapse
ExecStartPre=/bin/podman pull docker.io/matrixdotorg/synapse:latest
ExecStart=/bin/podman run
--name synapse
--volume /var/pods/synapse:/data:Z
-e SYNAPSE_SERVER_NAME=domain.com
-e SYNAPSE_REPORT_STATS=yes
--net virt_net
--ip 192.168.122.26
--mac-address 52:54:C0:A8:7A:1a docker.io/matrixdotorg/synapse:latest
ExecStop=/bin/podman stop synapse

[Install]
WantedBy=multi-use
```

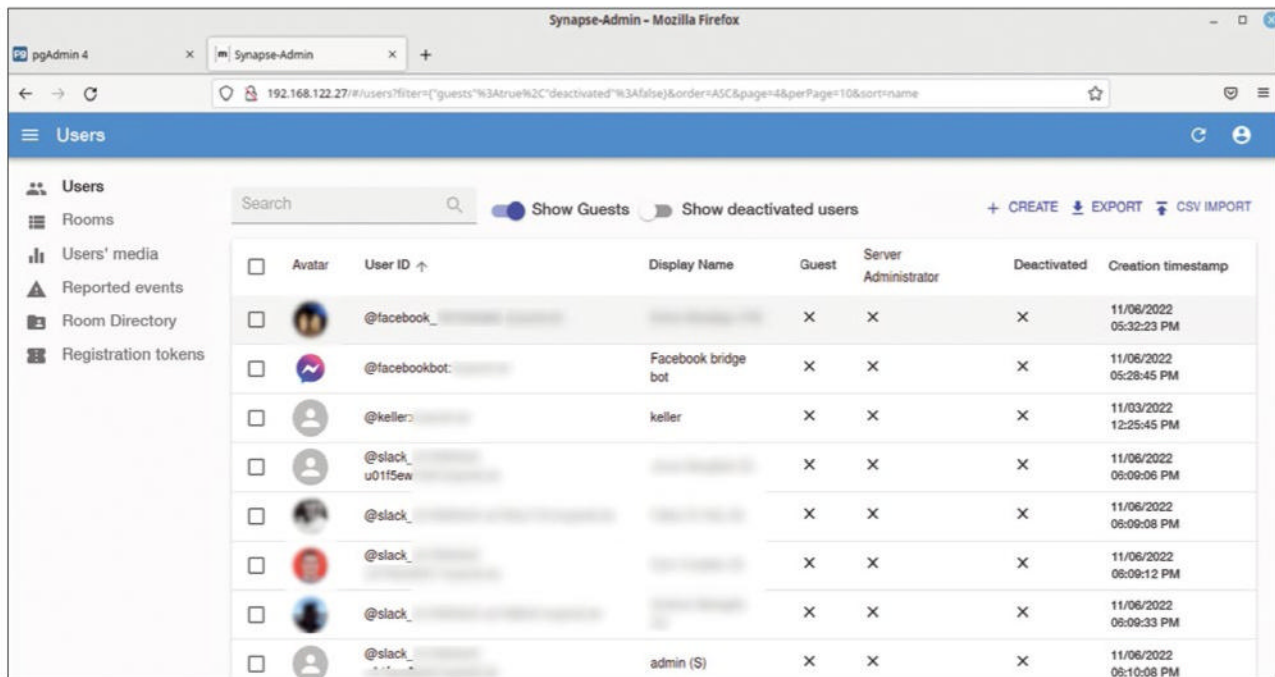



Figure 1: Synapse-admin simplifies user management. Bridges that connect to other chat networks automatically generate matching users.

admin UI projects. These work, but with limitations. The reverse proxy configuration provided above only exposes the messaging API (`/_matrix` and `/_synapse/client`) to the Internet, but not the admin API (`/_synapse/admin`), and it is a good idea to keep things that way. However, in this example, you could launch an admin web UI (Figure 1) directly on the hosting server in another container:

```
podman run 2
--net virt_net 2
--ip 192.168.122.27 2
--mac-address 52:54:C0:A8:7A:1b 2
docker.io/awesometechnologies/2
synapse-admin
```

Directly on the hosting server, call the URL `http://192.168.122.27` in the browser. In the connection dialog that follows, enter the admin account and the Matrix URL to match the `http://192.168.122.26:8008` example because you are using the internal network connection between the containers and are not routing via the proxy. If your hosting server doesn't have a GUI, you can tunnel the application port of the web UI `192.168.122.27:80` to your client in the usual way over SSH and then use the browser there.

Checking Federation

At this point, your Synapse server is running and the locally created users can communicate through it. Now it is time to test the connection to other matrix domains. First, check whether the federation settings of your server are configured correctly. To do so, simply go to the Matrix Federation Tester [2] and enter your domain name. The federation tester checks both discovery methods over DNS and the well-known REST API. The service also validates the HTTPS certificates. If everything shows up in green, other Matrix users can reach your server, and your users are allowed to communicate with other domains in return. Matrix manages all chats in rooms. Private chats for Matrix are nothing more than groups with only two members. Communication on the local server and with other Matrix installations is protected by end-to-end encryption. The messages themselves are saved in an encrypted format by the Synapse setup in the specified database.

Building Bridges

A Matrix setup with Synapse as the server and Element as the universal

client is sufficient for secure internal communication and chats with other Matrix users. However, most users use other chat platforms – first and foremost, WhatsApp. One of the great strengths of Matrix is its bridges. These plugins let Synapse connect to other services like WhatsApp, Google Chat, Slack, Discord, and Telegram with bots (Figure 2). With a comprehensive Synapse setup, users only need a single Matrix client to handle all chats, regardless of the platform. The bridge setup is quite similar for many plugins, but I will look at what is probably the most important bridge as a representative example here: WhatsApp.

The bridge runs as a standalone application, and it can theoretically run on a completely different computer than Synapse. In the setup discussed here, the *mautrix-whatsapp* bridge runs as a separate container with its own IP address (`192.168.122.31`) on the same system as Synapse. The plugin needs its own directory and database. As with Synapse, SQLite is fine for a trial run, but larger setups should use PostgreSQL. Create the `/var/pods/mautrix-whatsapp` directory and start the plugin container:

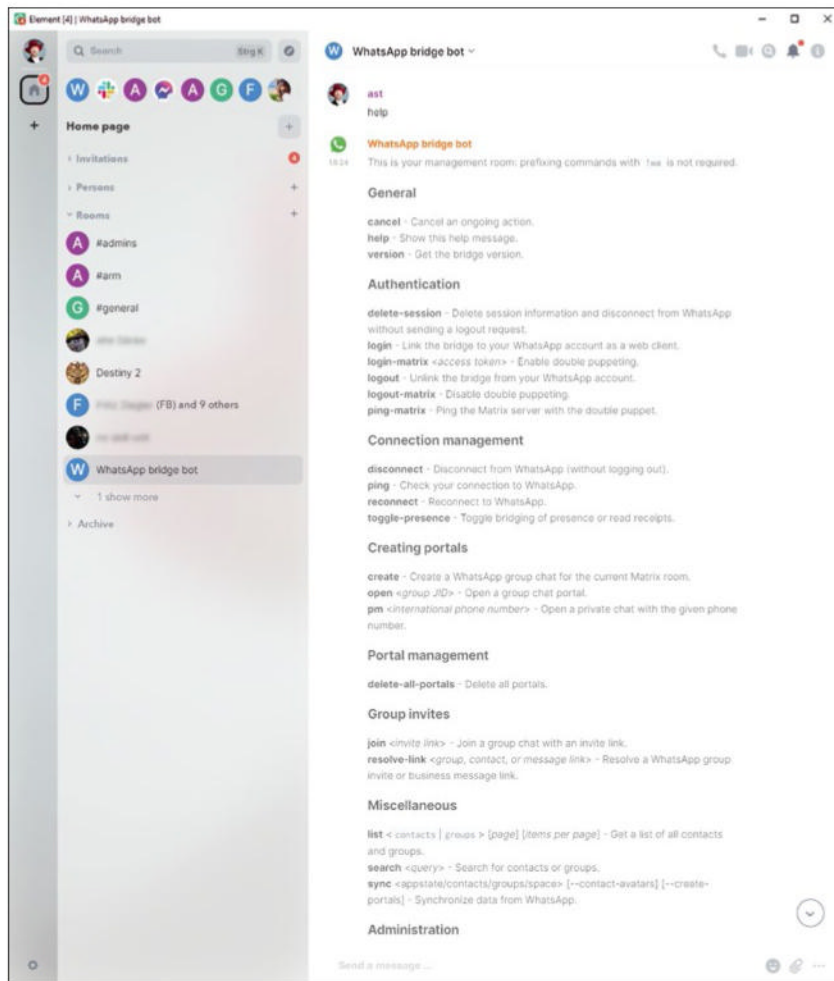


Figure 2: The Element desktop lists Matrix users and rooms, as well as Facebook, WhatsApp, and Slack users and groups. Users can control the functions of the bridges with the bridge bot.

```
podman run 2
--rm --name maatrix_whatsapp 2
--volume /var/pods/maatrix_whatsapp:2
/data:Z dock.mau.dev/maatrix2
/whatsapp:latest
```

When first launched, the bridge does not detect a configuration file. The container therefore creates a `config.yaml` with default settings in the specified directory and stops; you need to customize this file for your installation. The main entries for this setup are:

```
homeserver:
  address: http://192.168.122.26:8080
  domain: domain.com
appservice:
  address: http://192.168.122.31:29318
  hostname: 0.0.0.0
  port: 29318
```

```
database:
  type: sqlite3
  uri: whatsapp.db
```

Additionally, you need to configure the permissions setting to match your domain so that only users from your domain are actively allowed to use the bridge:

```
permissions:
  "": relay
  "domain.com": user
  "@admin:domain.com": admin
```

The plugin addresses the Synapse server directly over the internal address and provides its own service on the internal IP address of the container in return. In contrast to what the plugin documentation states, the database URL for SQLite

contains only the filename of the database without the path and without `sqlite://` at the beginning. The next time you start the container, include the IP information:

```
--net virt_net --ip 192.168.122.31 2
--mac-address 52:54:00:A8:7A:1f
```

Again, if you are running your containers without a bridge network, set all addresses to `127.0.0.1`, and instead of the network information, enter only the port `-p 29318:29318`.

When called for a second time, the plugin creates the database and checks the connection to the Synapse server. If the setup is OK, the *maatrix-whatsapp* bridge generates a `registration.yaml` and stops the container. This registration contains the plugin's access credentials for the Synapse server. Create a copy of the `registration.yaml` file in the Synapse server directory, preferably with the plugin name; that is:

```
cp /var/pods/maatrix_whatsapp/2
  registration.yaml 2
/var/pods/synapse/2
  registration_whatsapp.yaml
```

Then, edit the configuration of the Synapse server in `/var/pods/synapse/homeserver.yaml` to include

```
app_service_config_files:
  - /data/registration_whatsapp.yaml
```

and restart the Synapse container. For each additional bridge plugin, you need to create a separate registration file and add it to the Synapse service in the same way. If you now start the *maatrix-whatsapp* container again (without `--rm`), it registers with the Synapse service and remains active as a bot from this point on. Other bridges (e.g., for Facebook Messenger or Slack) use an almost identical approach up to this point. However, the method of authenticating against a particular service will differ.

If you want to use the WhatsApp bridge, you need two things: a Matrix client like Element (preferably on the

desktop) and your cell phone with the WhatsApp client. In Matrix, invite the WhatsApp bot `@whatsappbot:domain.com` to a chat. In the private chat room enter the commands for the bot – first of all, `login`. In response, the bot now sends a QR code. In the WhatsApp mobile app, go to *Linked devices*, and when you get there, select *Add Device*. Scan the QR code to allow the WhatsApp bot to access the chats; then, give the bot some time to read your WhatsApp configuration. By the way, the bot lists all functions if you type the `help` command.

The bot does not sync all your WhatsApp communication directly with Matrix. Instead, you can

individually choose which private WhatsApp chats and rooms you want to synchronize. To add a group to Matrix, first list the available groups with the `list groups` bot command. The bot returns the list and a unique ID for each group. If you now enter `open <group-ID>` as the command, the bot will create a Matrix chat room to match the group. Everything you post there from your Matrix client will appear in WhatsApp exactly as if you were typing it directly into the WhatsApp client. You can create private chats with the `pm <phone number>` bot command to communicate with Matrix and WhatsApp users from the Element client. Of course, WhatsApp

communication is then without the end-to-end encryption.

Conclusions

The basic setup of a chat infrastructure with Synapse takes some planning and time. In return, you can look forward to a secure communications platform beyond the established third-party providers. Thanks to the many bridges, Matrix allows communication with the popular chat networks. ■

Info

[1] Matrix: [\[https://matrix.org\]](https://matrix.org)

[2] Matrix Federation Tester: [\[https://federationtester.matrix.org/\]](https://federationtester.matrix.org/)

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?

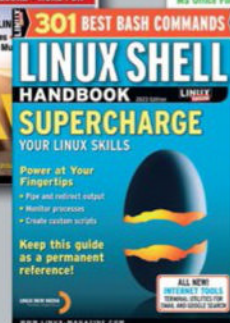
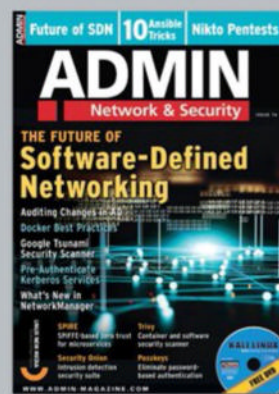
➤ sparkhaus-shop.com




➔ sparkhaus-shop.com

DIGITAL & PRINT
SUBSCRIPTIONS

SPECIAL EDITIONS





Secure collaboration

Productivity Storm

Sandstorm lets you self-host web-based productivity apps, apply individual permissions, and isolate documents for security with no effect on productivity. By Matthias Wübbeling

Sharing files is an important topic in team productivity. All employees need reliable access to required information for successful collaboration. Sandstorm is a security-hardened web app package manager built by a community of volunteers to run open source web applications [1]. Sandstorm's server-side sandboxing lets you isolate documents securely with little to no effect on productivity.

Security Risks in Modern Collaboration

The trend in IT has been toward microservices. Ever since hardware virtualization became widespread, individual services have run separately on different virtual machines. Although hard disk space has always been comparatively affordable,

virtualization comes at the price of memory overhead for a full-fledged operating system that gives you access to the physical resources of the computer through paravirtualized drivers.

Modern platforms with container technology, such as Kubernetes, further optimize resource consumption, especially in terms of memory consumption for microservices. Namespaces in the Linux kernel mean that it is no longer necessary to provide an operating system to isolate a piece of software from other running programs or specific files on the filesystem. It is solely a matter of a program's immediate runtime environment (i.e., the shared dynamic system libraries). Calls to the program and library kernel functions can even be processed by a single kernel.

The architecture described here leads to each individual application (e.g., software for cooperative document editing or calendar systems) running in its own container without direct access to the resources of other processes. A database connected on the back end also runs in its own environment, and communication then takes place over a private network that is virtualized in the kernel. Standard tools have been developed for communication over the network. In terms of IT security, this structure might be unsuitable, no matter how elegant it appears at first glance. Suppose an external user gains access to a company's filesharing service because they need to exchange data with one of the internal employees. A potential attacker who gains access to this user account then controls an

Photo by Mateusz Klein on Unsplash

authenticated system user. To be able to access further documents, the attacker only needs to find a vulnerability that allows privilege escalation. The developers of Sandstorm take things a step further: Instead of isolating individual services in individual containers, the software isolates the information itself along with the associated applications. Therefore, every document a user creates in Etherpad, for example, is also assigned its own Etherpad instance and, in an ideal scenario, is the only document in that Etherpad.

From now on, access management to this Etherpad instance is handled by the Sandstorm platform, which can distinguish between read, comment, and read/write permissions. Conceptually, Sandstorm basically demarcates available apps from the documents created with them – the “grains” in tech-speak – to achieve granular access control in the truest sense of the word.

Installing Sandstorm

The hardware requirements for installing Sandstorm are initially low – at least on paper. The developers expect a recent 64-bit Linux with at least 1GB of RAM. The required hard disk size is not specified; Sandstorm itself takes up just north of 2.8GB. In the lab I used, I discovered that a machine with 2GB of RAM and 20GB of storage was fine.

Depending on your choice of installation method, the process can be anything from very simple to pretty complex. You can choose anything from a script, right up to building Sandstorm yourself. I’ll take the simplest route here and start the installation directly on the console of an Ubuntu server with the provided script:

```
curl https://install.sandstorm.io | bash
```

During the process, you will need to answer prompts about your installation. To test Sandstorm, select a standard installation in the first prompt by pressing Enter. After a short look at the installation steps, you also

confirm the next prompt. If you are already running an HTTP server on port 80 or 443 or an email server on port 25, the script will suggest alternative ports, which you can adapt to your local conditions.

The Sandstorm developers have also put some thought into protecting your instance with the Transport Layer Security (TLS) protocol. The `sandcats.io` domain has free subdomains to help you secure your setup with a certificate from Let’s Encrypt. Choose a creative name (some obvious subdomains are already taken) and register with an email address. You do not need to confirm this, but make sure you use an existing address to be able to recover your chosen subdomain later on.

After downloading and installing Sandstorm, you again need to enter an email address at the command line. This time, you need it to register with Let’s Encrypt, and you can use a different address than in the previous step. After a short wait for the TLS certificate to install, you will see a link onscreen, which means that the installation completed successfully. Use the link to log in as admin to the instance you just created (i.e., open the link in your browser).

Configuration

You can now start configuring your own instance in the web browser by

pressing the matching button. The first step is to select at least one authentication method that you want to offer to your users (Figure 1). The simplest option is passwordless authentication by email. To do this, you need to enter an existing email account, and Sandstorm will then mail a corresponding link whenever you log in. If you use an LDAP server in your organization, you can add its parameters. Alternatively, choose a preconfigured OAuth provider with Google or GitHub or configure your own services with OpenID Connect or Security Assertion Markup Language (SAML). Of course, you can also enable multiple back ends at the same time.

In the second step for configuring your organization’s settings, you need to define a domain for email authentication (i.e., so you can check that it belongs to your organization). You can also restrict the ability to share documents with external guests (Figure 2). A login to your organization would then always be required to access documents. The previously selected option to store all users of an organization in the address book makes it easier for your users to share with each other but also exposes the list of active users to everyone.

After clicking *Save and continue*, you are taken to the email setup. Sandstorm sends email to users for various

Login provider	Status	
E-mail (passwordless)	Not enabled	Configure
Google	Not enabled	Configure
GitHub	Not enabled	Configure
LDAP	Not enabled	Configure
OpenID Connect	Not enabled	Configure
SAML	Not enabled	Configure

Figure 1: Select login options that are the right fit for your organization.

reasons, not only to log in by email. To do this, they need an account on the mail server that allows sending with different domain addresses. Sandstorm sometimes communicates with different sender addresses; for example, each app has its own address for email. The use of a free mail provider is difficult. Once you have stored a valid email account, the apps selected by the developers will be pre-installed in the next step. Because this has already happened in the background, you can probably click *Next* here. You can customize the list of apps pre-installed for users later in the Admin panel. At the end of the configuration process, you will need to create an account with admin rights. To do this, enter the email address in the appropriate form, press the button to send a login email, and press *Next* to complete the configuration.

Installing Apps

Once you have logged in to the Sandstorm instance with your account, you will see a very straightforward menu on the left where you can select options. After logging in, you are taken to the Apps view. You can get started directly or press the large square with the plus sign (+), which opens the App market, and you can browse through the available apps at your leisure. The examples chosen for this article are the GitWeb and EtherCalc collaboration tools. GitWeb supports collaboration by Git and lets you share maintenance of Git web pages. EtherCalc is a tool for collaborative spreadsheet editing. After selecting an app in the Market, the app is first downloaded to the server. Next, press *Install* in the Sandstorm dialog once again to add the app to your user account.

GitWeb Example

After installing GitWeb, create a new repository at the bottom of the display. In the Grain view that now opens, you can see the commands in the text box that will let you clone Git at the command line, so run the commands on the console and change to the newly created directory.

In addition to the classic Git functions, GitWeb offers its own web server. You can create content for it in the `gw-pages` branch; run the commands

```
git checkout -b gw-pages
$ echo "<h1>Hi there</h1>" > index.html
$ git add index.html
$ git commit -m "first commit"
$ git push origin gw-pages
```

to create this branch and add initial content for your website. The server message during the push gives you the URL where your new website can now be found. The web pages are automatically deployed to this branch on the web server with each push. Now return to the Sandstorm web interface to share the repository you just created with your fellow workers. Select *Share access* at the top of the web page and choose one of two options: Invite users by email address or create a link that you can share. In both cases you need to select the desired access permissions, either read-only (*CAN READ*) or read and write (*CAN READ AND WRITE*).

Regardless of whether you have the system send an email or send the link you generated yourself, a share link for the grain remains valid until the grain or the share itself is deleted. An overview of all shares for a grain can be found in the Share Access dialog at bottom right after the *See who has access* link.

If the invited person now follows the given link, they gain access to one grain. Without an active login (i.e., in anonymous edit mode), the menu on the left side is missing. The view of GitWeb itself and the description for cloning the repository are the same

Figure 2: In the second step of the installation, you enable authentication with the email domain.

as those seen by the owner of the grain, but the credentials used for the repository are different.

EtherCalc

Clicking the *Create new spreadsheet* link in the EtherCalc app display opens the spreadsheet with the new document. The Sandstorm frame around the document does not change. In the menu on the left-hand side, you see both the GitWeb grain and spreadsheet, which is now also listed for direct access.

To avoid losing track, you can change the names of the grains if you have several open by clicking *Untitled EtherCalc spreadsheet* at the top. If you use *Share access* to share the document after the name change, it will be shown to the other users with the new name, too. Name changes by anonymous users are accepted by the Sandstorm interface but are not saved or synchronized. If a logged-in user changes the name of a grain, this name is stored locally but not synchronized. Instead, the name used by the owner is displayed, as well. If you want to revoke a previously shared link or the associated access to your grain, selectively delete both in the Shares overview. Unfortunately, Sandstorm does not display the link itself, but only the label you specified. You therefore have to rely on your own descriptions when deleting the links, making it essential to specify a meaningful label when you create something. After deletion, the grains in the link are no longer available; sessions opened at the same time are immediately deleted and the user will see a message to this effect.

Workaround for Creating Groups

Sandstorm does not provide for the possibility of dividing people into groups, apart from the members of the organization who are identified by the domain of the email address.

However, grains can be organized into groups in the Collections app by combining different documents in a group and then sharing them with a single link. You can set individual permissions for each document in this collection. Groups are convenient if you want to share multiple documents with multiple editors. Even after sharing, additional grains can be added to a group so that you only need one collection for teams, which then contains all documents. In fact, you can add more collections to a collection, even recursively. However, avoid overdoing recursion, especially with different permissions; in our tests, this practice repeatedly led to session errors being thrown.

Useful Management Tools

In addition to the functions already discussed, Sandstorm offers useful features for managing grains, which you can access from the menu at the top of the page. The *Move to trash* feature, as expected, deletes the document and all associated shares and resources. The *Show Debug Log* display icon points experts and developers to log messages to assist with troubleshooting in case of problems. However, as a normal user, you won't find much information for working with grains here. The *Restart App* icon should also not be used under normal circumstances.

The *Download Backup* option for backing up a grain, on the other hand, is very useful. The icon downloads a ZIP file that you can use for recovery, which can be triggered by pressing the *Grains | Restore backup* button at top right. If the original grain still exists for your user, Sandstorm creates a new instance with the status at the time of the backup. Unfortunately, the shares are not part of the backup, so you have to create them again manually after a restore. If you want to test different scenarios within a grain or need different

versions, the clone function is a useful tool. Much like restoring a backup, Sandstorm duplicates the grain in its current state. In this case, as with a backup, shares are not transferred to the cloned object. You would need to create these again if needed.

Conclusions

The list of apps you can use in Sandstorm is already considerable and is growing. Setting up the software is as easy as pie for administrators. Although the application starts the apps multiple times – in addition to the grains after access – resource usage is fairly low compared with other container environments. Sandstorm is therefore useful for running on less powerful machines. Unfortunately, user and grain management are currently fairly rudimentary. The assignment to your organization is based entirely on your choice of the email domain, and you can only create groups with the Collection app. All told, though, Sandstorm offers a carefully considered and surprisingly functional approach to isolating documents in an anonymously shareable and cooperatively usable environment. ■

Info

[1] Sandstorm: [\[https://sandstorm.io\]](https://sandstorm.io)

The Author

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



Detect failures and ensure high availability

On the Safe Side

Eliminate single points of failure and service downtime with the DRBD distributed replicated storage system and the Corosync and Pacemaker service. By Petros Koutoupis

Many clustering and high-availability frameworks have been developed for Linux, but in this article, I focus on the more mainstream and widely used Corosync and Pacemaker service and DRBD. If you follow along, you'll learn how to configure an active-passive dual-node cluster that replicates local storage (accessed and written to/read from vital applications or services) to its neighboring node. In this way, you'll be able to host and serve data requests, so long as a single node of the cluster remains online.

High Availability

As we depend more and more on technology and the services it provides, availability becomes increasingly important. In the recent past, hardware vendors and solution providers charged a lot of money for

proprietary products that ensured a high level of tolerance for hardware and I/O path failures. Those days have come and gone. Today, the data center has evolved, and with that evolution comes the adoption of more commodity hardware (i.e., hardware that is not built with the level of resilience or sophistication typically seen on a mainframe, an IBM POWER processor, or other equivalent machines). These non-commodity machines were designed from the bottom up to sustain all sorts of internal hardware and path failures – redundant memory, CPUs, network interfaces, power supplies, and more. However, that type of functionality also came at a much higher cost. A level of complexity also was introduced by those same proprietary systems. Diagnosing, replacing, and repairing faulty or problematic components required

both deep pockets and a well-trained technician. These factors were, and continue to be, the primary reasons for opting for commodity technology (containing only a subset of hardware redundancy) and instead relying on the software to handle all sorts of failure scenarios.

More affordable off-the-shelf server solutions provided the data center with the flexibility to build an ecosystem in the preferred way. The only limitation was the extent of the administrators' imaginations. What pieced all of these moving parts together was the software – the very same software that filled the void in fault tolerance.

Therefore, it shouldn't be surprising that many open source software solutions are doing exactly that. Production-grade open source projects have evolved and matured enough to stay

Photo by Tim Foster on Unsplash

competitive with the proprietary solutions of yesterday, and they provide the same feature-rich solutions, if not more.

Zero Downtime

Failures will happen, both in hardware and software. Even if that off-the-shelf commodity server does provide redundant power supplies or its local storage is configured in a RAID configuration, other things can and will go wrong. Processors can fail. Memory can go bad and corrupt vital data. Error correction won't always detect and correct every faulty bit. To address these pain points, systems need to be made redundant by enabling multiple components to perform the same set of tasks. Sometimes you can do this by setting up more than one similar machine and configuring them to be highly available and to accomplish the same set of functions.

The idea behind high availability (HA) is simple: Eliminate any and all single points of failure to ensure that, if a server node or communication path to the underlying storage or service goes down, data requests can still be served. The ultimate goal of configuring a high-availability ecosystem is to provide continuous and uninterrupted service for sometimes critical business applications, all while masking both planned and unplanned outages, including failures that can be a result of system crashes, network failures, storage issues, and more.

Downtime can cost a company time and resources and potentially a loss in business. It's necessary to identify any and all single points of failure and eliminate them by configuring redundant instances, sometimes even balancing the workload across those same redundant instances through a concept typically referred to as "multipath." High-availability technologies are designed to detect failures automatically and recover from them immediately.

High availability doesn't guarantee zero downtime, but you can get pretty close

to it. If configured appropriately, you can aim for 99.999 percent uptime.

High Availability Concepts

In the software realm, fault tolerance takes on a unique role. To ensure a level of redundancy, I want to cover a few configuration concepts:

- **Active-Active:** In an active-active configuration, services and resources are accessible from any and all nodes within the cluster simultaneously. If one node fails, it will not affect availability of the
- same service or resources to the rest of the nodes.
- **Active-Passive:** In an active-passive configuration, services and resources are available from only a single node at a time (Figure 1). The rest of the nodes remain passive for that particular service or resource. In the event of a failure on a node hosting that resource, one of the passive nodes resumes availability of that resource.
- **Failover (and Failback):** Failover is when a service or availability of a resource fails over from one node

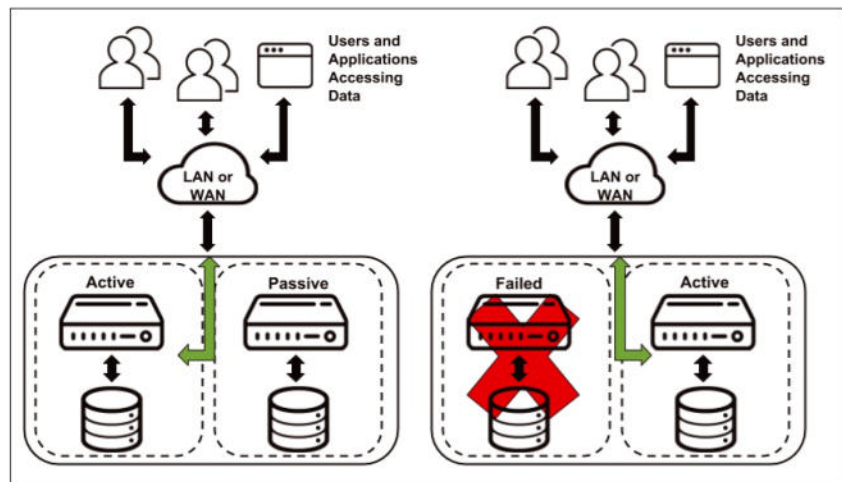


Figure 1: In an active-passive configuration, only one node is actively serving data requests in both healthy and failed states.

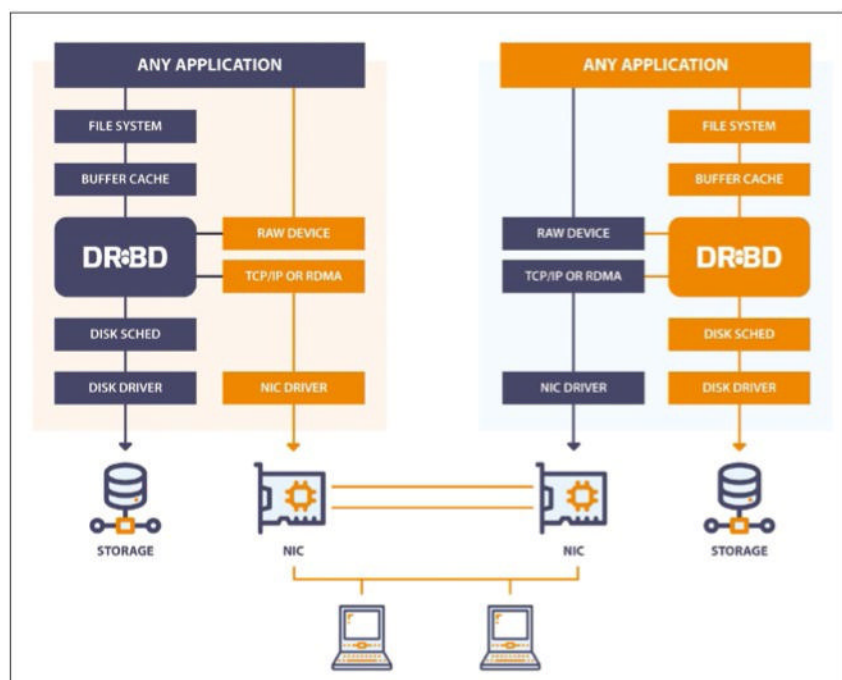


Figure 2: A high-level layout of the DRBD architecture and design showing where DRBD sits in the Linux kernel storage stack. © Courtesy of LINBIT

in the cluster to another. Sometimes when the failed node is back online and in a healthy state, that same service can and will fail back to its original node if configured. Note that the passive nodes don't need to stay idle. If configured accordingly, resources and services can be balanced across all nodes within the cluster. For instance, Node 1 can host Resource 1, while Node 2 hosts Resource 2. If Node 1 fails, then Node 2 would host both Resource 1 and Resource 2.

Corosync and Pacemaker

Corosync has become the standard bearer of everything cluster-related

in Linux. Almost all modern Linux distributions support the framework, which also is available in their respective package repositories. Corosync is a cluster engine, and more technically, it's a communication system or messaging layer. Corosync's purpose is to monitor both cluster state and quorum. Pacemaker is a cluster resource manager. When you enable a particular resource agent catering to a specific set of tasks or applications, Pacemaker ensures that it's running on the node on which it's designated to run. If that node fails, Pacemaker redirects that resource over to another node within the same cluster.

Listing 1: Test Ping

```
petros@ubu22042-1:~$ ping ubu22042-2
PING ubu22042-2 (10.0.0.62) 56(84) bytes of data.
64 bytes from ubu22042-2 (10.0.0.62): icmp_seq=1 ttl=64 time=0.505 ms
64 bytes from ubu22042-2 (10.0.0.62): icmp_seq=2 ttl=64 time=0.524 ms
64 bytes from ubu22042-2 (10.0.0.62): icmp_seq=3 ttl=64 time=0.496 ms
^C
--- ubu22042-2 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2312ms
rtt min/avg/max/mdev = 0.496/0.508/0.524/0.011 ms
```

Listing 2: DRBD Config Details

```
# You can find an example in /usr/share/doc/drbd.../drbd.conf.example

include "drbd.d/global_common.conf";
include "drbd.d/*.res";
resource r0 {
    protocol C;
    startup {
        wfc-timeout 15;
        degr-wfc-timeout 60;
    }
    net {
        cram-hmac-alg sha1;
        shared-secret "secret";
    }
    on ubu22042-1 {
        device /dev/drbd0;
        disk /dev/sdb;
        address 10.0.0.216:7788;
        meta-disk internal;
    }
    on ubu22042-2 {
        device /dev/drbd0;
        disk /dev/sdb;
        address 10.0.0.62:7788;
        meta-disk internal;
    }
}
```

DRBD

DRBD (distributed replicated block device) [1] is implemented as both a kernel module and userspace management applications. It's developed by LINBIT, and as the name implies, the sole purpose of this framework is to replicate block devices across multiple distributed machines configured in a cluster. This method of replication is similar, but not identical, to that of a RAID 1 mirror (Figure 2). DRBD is typically configured for a high-availability environment, wherein one server acts as the primary server, making its underlying DRBD block device available to a specific set of services or applications, while all its updates are distributed across the cluster. The volumes on the remaining servers are always kept consistent with the primary server.

Configuring Servers

In this example, I configure a cluster consisting of two servers hosting data. When the data is updated to the one server, under the hood, DRBD will replicate it to the secondary server and its block storage. This scenario will be ideal when services such as NFS are hosted and the primary node becomes unresponsive or fails. Access to the necessary file-sharing services and its file content will continue uninterrupted. Both hosts have the exact same configuration, and I will be using a secondary (local) drive with the exact same capacity on each, /dev/sdb:

```
$ cat /proc/partitions|grep sd
      8      0  10485760 sda
      8      1    1024 sda1
      8      2  1835008 sda2
      8      3   8647680 sda3
      8     16 20971520 sdb
```

It is extremely important that both nodes in the cluster see one another in the network and can access the companion node by its hostname. If not, then be sure to modify your /etc/hosts file to address that requirement. If you are not sure about whether your hosts can reach each other, a simple ping or

ssh from one to the other should provide the answer (Listing 1). Also note in this example that my command-line commands reflect that I am using Ubuntu as the host operating system, although DRBD is not limited to Ubuntu. The operating system of your choice will likely look different for the following package management examples. Before proceeding, be sure to update your operating system on both nodes to the latest and greatest package revisions to ensure that you are running in a stable and secure environment:

```
$ sudo apt-get update &&
sudo apt-get upgrade
```

Again, on both nodes, install the DRBD package from the operating system's remote repository:

```
$ sudo aptitude install drbd-utils
```

On one node, open the `/etc/drbd.conf` file and add the details in Listing 2, making the proper adjustments to account for hostnames, IP addresses, and disk drives.

Copy the configuration file over to the secondary server and run the final set of commands on both servers to create the metadata block, start the DRBD service daemon, and enable that same service to start on every bootup:

```
$ sudo scp /etc/drbd.conf ubu22042-2:/etc/
$ sudo drbdadm create-md r0
initializing activity log
initializing bitmap (640 KB) to all zero
Writing meta data...
New drbd meta data block successfully
created.
$ sudo systemctl start drbd.service
$ sudo systemctl enable drbd.service
```

You should now see a new block device listed on each machine:

```
$ cat /proc/partitions|grep drbd
147      0   20970844 drbd0
```

Next, configure the primary (active) node of the storage cluster:

```
$ sudo drbdadm --
--overwrite-data-of-peer primary all
```

You will not be able to run this command more than once or on more than one node; otherwise, you'll see a message like:

```
$ sudo drbdadm --
--overwrite-data-of-peer primary all
0: State change failed: (-1) Multiple
primaries not allowed by config
Command 'drbdsetup-84 primary 0
--overwrite-data-of-peer' terminated
with exit code 11
```

At this point, the `/dev/drbd0` of the primary node will synchronize over to the second. To watch the synchronization progress from the primary node to the secondary, on the secondary node, enter:

```
$ watch -n1 cat /proc/drbd
```

You should see something similar to the output shown in Listing 3. When synchronization is 100 percent complete, the same file will showcase the output in Listing 4.

At any point you can switch both primary and secondary roles by going to the primary node and executing:

```
$ sudo drbdadm secondary r0
```

(Note: The block device must not be mounted.) Going to the secondary node, enter:

```
$ sudo drbdadm primary r0
```

To validate, read the content of `/proc/drbd` and observe the order of the Primary and Secondary labels in the *Connected ro* section:

```
0: cs:Connected ro:Secondary/Primary
ds:UpToDate/UpToDate C r-----
```

Back on the primary node, format the DRBD storage volume with a filesystem, mount the DRBD storage volume, and verify that it's mounted:

```
$ sudo mkfs.ext4 -F /dev/drbd0
$ sudo mount /dev/drbd0 /srv
$ df|grep drbd
/dev/drbd0    20465580    24
19400632    1% /srv
```

You won't be able to mount the block device on the secondary node, even when fully synchronized. If you do, you'll see the error:

```
$ sudo mount /dev/drbd0 /srv
mount: /srv: mount(2) system call
failed: Wrong medium type.
```

Back on the primary node, create a test file:

```
$ echo "hello world" |
sudo tee -a /srv/hello.txt
hello world
$ cat /srv/hello.txt
hello world
$ ls -l /srv/hello.txt
-rw-r--r-- 1 root root
12 Apr  8 16:37 /srv/hello.txt
```

Listing 3: Sync Progress

```
Every 1.0s: cat /proc/drbd
ubu22042-2: Sat Apr  8 16:19:01 2023
version: 8.4.11 (api:1/proto:86-101)
srcversion: 2A5DFCD31AE4EBF93C0E357
0: cs:SyncTarget ro:Secondary/Primary ds:Inconsistent/UpToDate C r-----
ns:0 nr:1755136 dw:1755136 dr:0 al:8 bm:0 lo:1 pe:4 ua:0 ap:0 ep:1 wo:f oos:19215708
[>.....] sync'ed: 8.4% (18764/20476)M
finish: 0:10:40 speed: 29,964 (16,712) want: 41,000 K/sec
```

Listing 4: cat /proc/drbd

```
$ cat /proc/drbd
version: 8.4.11 (api:1/proto:86-101)
srcversion: 2A5DFCD31AE4EBF93C0E357
0: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r-----
ns:0 nr:20970844 dw:20970844 dr:0 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

Assuming you wait the proper amount of time for the file to synchronize over to the secondary node, unmount the DRBD volume on the primary node and demote the node to a secondary role:

```
$ sudo umount /srv
$ sudo drbdadm secondary r0
```

On the secondary node, promote its role to that of a primary, and then mount the DRBD volume:

Listing 5: /etc/corosync/corosync.conf

```
totem {
    version: 2
    cluster_name: drbd-cluster
    crypto_cipher: none
    crypto_hash: none
    transport: udpu
    interface {
        ringnumber: 0
        bindnetaddr: 10.0.0.62
        broadcast: yes
        mcastport: 5405
    }
}

quorum {
    provider: corosync_votequorum
    two_node: 1
}

logging {
    to_logfile: yes
    logfile: /var/log/corosync/corosync.log
    to_syslog: yes
    timestamp: on
}

nodelist {
    node {
        ring0_addr: 10.0.0.216
        name: ubu22042-1
        nodeid: 1
    }
    node {
        ring0_addr: 10.0.0.62
        name: ubu22042-2
        nodeid: 2
    }
}

service {
    name: pacemaker
    ver: 0
}
```

```
$ sudo drbdadm primary r0
$ sudo mount /dev/drbd0 /srv
```

Now you should be able to see the same test file created earlier:

```
$ ls -l /srv/
total 20
-rw-r--r-- 1 root root 20 12 Apr  8 16:37 hello.txt
drwx----- 2 root root 16384 Apr  8 16:33 lost+found
$ cat /srv/hello.txt
hello world
```

Note that if you are seeing the following error when changing node roles,

```
$ sudo drbdadm secondary r0
0: State change failed: (-12) Device is held open by someone
Command 'drbdsetup-84 secondary 0' terminated with exit code 11
```

you can determine which process ID is holding the device open by typing:

```
$ sudo fuser -m /dev/drbd0
/dev/drbd0: 597
```

In my case, it is the multipathd daemon:

```
$ cat /proc/597/cmdline
multipathd
```

After disabling the service:

```
$ sudo systemctl stop multipathd
Warning: Stopping multipathd.service, but it can still be activated by: multipathd.socket
```

I had no problem continuing with my exercise.

Installing and Configuring Corosync/Pacemaker

On both nodes, install the following Corosync and Pacemaker packages (all required dependencies will be installed):

```
$ sudo aptitude install pacemaker crmsh
```

On the primary node, generate a Corosync cluster key:

```
$ sudo corosync-keygen
Corosync Cluster Engine Authentication key generator.
Gathering 2048 bits for key from /dev/urandom.
Writing corosync key to /etc/corosync/authkey.
```

Listing 6: Cluster Status

```
$ sudo crm status
Cluster Summary:
* Stack: corosync
* Current DC: ubu22042-2 (version 2.0.3-4b1f869f0f) - partition with quorum
* Last updated: Sat Apr  8 17:35:53 2023
* Last change: Sat Apr  8 17:35:10 2023 by hacluster via crmd on ubu22042-2
* 2 nodes configured
* 0 resource instances configured
Node List:
* Online: [ ubu22042-1 ubu22042-2 ]
Full List of Resources:
* No resources
```

Listing 7: Properties

```
$ sudo crm configure show
node 1: ubu22042-1
node 2: ubu22042-2
property cib-bootstrap-options: have-watchdog=false dc-version=2.0.3-4b1f869f0f
cluster-infrastructure=corosync cluster-name=drbd-cluster stonith-enabled=false
no-quorum-policy=ignore
```


Copy the newly created key over to the secondary node:

```
$ sudo scp /etc/corosync/authkey 2
ubu22042-2:/etc/corosync/
```

On both nodes, change the permissions to read-only for the file owner (root):

```
$ chmod 400 /etc/corosync/authkey
```

Again, on both nodes, modify `/etc/corosync/corosync.conf` to look like [Listing 5](#), making the proper modifications to the hostnames and IP addresses. The `bindnetaddr` field should be unique on each node; in this case, it should display the IP address of the server hosting the file.

On both nodes, enable and start both Corosync and Pacemaker services:

```
$ sudo systemctl enable corosync
$ sudo systemctl start corosync
$ sudo systemctl enable pacemaker
$ sudo systemctl start pacemaker
```

The cluster should now be up and running. On any node, list the status of the cluster ([Listing 6](#)). You should observe both nodes listed with an *Online* status.

For the purposes of this experiment, disable the STONITH (Shoot The Other Node In The Head) fencing technique and ignore the quorum state of the cluster:

```
$ sudo crm configure property 2
stonith-enabled=false
$ sudo crm configure property 2
no-quorum-policy=ignore
```

Now verify that these properties were set ([Listing 7](#)).

Fortunately, the DRBD project has been around for long enough that resource agents exist to manage the DRBD volumes. To list the Open Cluster Framework (OCF) DRBD-defined resource agents (for Pacemaker), enter:

```
$ crm_resource 2
--list-agents ocf|grep -i drbd
drbd
drbd.sh
```

To finish the cluster configuration, enter the Cluster Resource Manager (CRM) interactive shell:

```
$ sudo crm configure
```

Next, enable the configurations in [Listing 8](#), making the proper modifications to account for the disk device, mount directory, and filesystem type. To commit the configuration, enter:

```
crm(live/ubu22042-1)configure# commit
```

Now verify the configuration ([Listing 9](#)), and quit the CRM shell:

```
crm(live/ubu22042-1)configure# quit
bye
```

The cluster should now know about and officially manage access to the DRBD volume created earlier

Listing 8: Enable Config

```
crm(live/ubu22042-1)configure# primitive drbd_res ocf:linbit:drbd params drbd_resource=r0 op monitor
interval=29s role=Master op monitor interval=31s role=Slave
crm(live/ubu22042-1)configure# ms drbd_master_slave drbd_res meta master-max=1 master-node-max=1
clone-max=2 clone-node-max=1 notify=true
crm(live/ubu22042-1)configure# primitive fs_res ocf:heartbeat:Filesystem params device=/dev/drbd0
directory=/srv fstype=ext4
crm(live/ubu22042-1)configure# colocation fs_drbd_colo INFINITY: fs_res drbd_master_slave:Master
crm(live/ubu22042-1)configure# order fs_after_drbd mandatory: drbd_master_slave:promote fs_res:start
```

Listing 9: Verify and Validate Config

```
crm(live/ubu22042-1)configure# show
node 1: ubu22042-1
node 2: ubu22042-2
primitive drbd_res ocf:linbit:drbd params drbd_resource=r0 op monitor interval=29s role=Master op
monitor interval=31s role=Slave
primitive fs_res Filesystem params device="/dev/drbd0" directory="/srv" fstype=ext4
ms drbd_master_slave drbd_res meta master-max=1 master-node-max=1 clone-max=2 clone-node-max=1
notify=true
order fs_after_drbd Mandatory: drbd_master_slave:promote fs_res:start
colocation fs_drbd_colo inf: fs_res drbd_master_slave:Master
property cib-bootstrap-options: have-watchdog=false dc-version=2.0.3-4b1f869f0f
cluster-infrastructure=corosync cluster-name=drbd-cluster stonith-enabled=false
no-quorum-policy=ignore
```

Listing 10: Cluster Status

```
$ sudo crm status
Cluster Summary:
* Stack: corosync
* Current DC: ubu22042-2 (version 2.0.3-4b1f869f0f) - partition with quorum
* Last updated: Sat Apr 8 17:45:40 2023
* Last change: Sat Apr 8 17:42:15 2023 by root via cibadmin on ubu22042-1
* 2 nodes configured
* 3 resource instances configured
Node List:
* Online: [ ubu22042-1 ubu22042-2 ]
Full List of Resources:
* Clone Set: drbd_master_slave [drbd_res] (promotable):
* Masters: [ ubu22042-1 ]
* Slaves: [ ubu22042-2 ]
* fs_res (ocf::heartbeat:Filesystem): Started ubu22042-1
```

([Listing 10](#)); it is enabled on the primary and now “Active” node. Executing the `df` or `mount` command will confirm that it is mounted at `/srv`. The secondary node remains idle and “Passive” until your primary node becomes unavailable. To test, shut down the primary node,

```
$ sudo shutdown -h now
Connection to 10.0.0.216 closed by 2
remote host.
Connection to 10.0.0.216 closed.
```

wait a bit (about 10), and dump the status of the cluster from the secondary node ([Listing 11](#)).

Notice now that the primary node is listed as being OFFLINE, and the secondary node has resumed hosting the “failed over” DRBD volume:

```
$ df | grep drbd
/dev/drbd0          20465580      28 2
19400628    1% /srv
```

If you list files again,

```
$ ls -l /srv/
total 20
-rw-r--r-- 1 root root    2
12 Apr  8 16:37 hello.txt
drwx----- 2 root root 16384 2
Apr  8 16:33 lost+found
```

you find the same test file you created before.

Summary

As mentioned earlier, the primary objective for enabling a highly available environment is to reduce both single points of failure and service downtime. You definitely can expand on the examples here to work with more supported applications and functions. To learn more about Pacemaker and the resource agents the framework supports, visit the resource agents section of the Linux-HA wiki [\[2\]](#).

Listing 11: Secondary Node Status

```
$ sudo crm status
Cluster Summary:
* Stack: corosync
* Current DC: ubu22042-2 (version 2.0.3-4b1f869f0f) - partition with quorum
* Last updated: Sat Apr  8 17:47:41 2023
* Last change: Sat Apr  8 17:42:15 2023 by root via cibadmin on ubu22042-1
* 2 nodes configured
* 3 resource instances configured
Node List:
* Online: [ ubu22042-2 ]
* OFFLINE: [ ubu22042-1 ]
Full List of Resources:
* Clone Set: drbd_master_slave [drbd_res] (promotable):
  * Masters: [ ubu22042-2 ]
  * Stopped: [ ubu22042-1 ]
* fs_res (ocf::heartbeat:Filesystem): Started ubu22042-2
```

Info

[\[1\]](#) DRBD: <https://linbit.com/drbd/>

[\[2\]](#) Linux-HA resource agents wiki page: http://www.linux-ha.org/wiki/Resource_Agents

Author

Petros Koutoupis is currently a senior performance software engineer at Cray (now HPE) for its Lustre High Performance File System division. He is also the creator and maintainer of the RapidDisk Project (www.rapiddisk.org). Petros has worked in the data storage industry for well over a decade and has helped pioneer the many technologies unleashed in the wild today.

An open source object storage solution

Object Lesson

We introduce the MinIO high-performance object store, its key features and applications, and some performance tips. By Artur Skura

The world of data storage has evolved over the years, and as the amount of data we generate and manage continues to grow, we need more efficient and scalable storage solutions. One such solution is object storage, which has become increasingly popular because of its flexibility, cost-effectiveness, and scalability. Object storage is a data storage architecture that stores data as objects, rather than as files in a hierarchical filesystem or blocks in a block storage system. Each object includes the data, metadata, and a unique identifier, allowing for easier access and management of large amounts of unstructured data.

In this article I introduce you to MinIO, a popular object storage solution. MinIO's source code is available under the GNU Affero General Public License v3.0, which means you can customize, extend, and contribute to the project.

Overview of MinIO

MinIO is a high-performance, open source, object storage system compatible with Amazon Simple Storage Service (S3) and designed for unstructured data (see the "Key Features" box). Developed with a focus on simplicity, scalability, and performance, MinIO is designed for

use in private, public, and hybrid cloud environments. MinIO supports a wide range of use cases, including big data analytics, machine learning (ML), backup, and archiving. In particular, MinIO's high throughput and low-latency performance make it well-suited for artificial intelligence (AI) and ML workloads. Data scientists around the world use MinIO to store large volumes of training data and serve it to ML models for training and inference.

Although the major public cloud platforms will tell you that going multicloud is an anti-pattern and will implement features to discourage you from using the services of the competition, in reality most large companies use more than one cloud. For example, they might use AWS for their core business, Azure for some of its features such as directory management, and Google Cloud Platform (GCP) for Kubernetes. MinIO's compatibility with the S3 API and support for multitenant environments make it an attractive option for hybrid cloud and multicloud storage deployments. Several companies use MinIO to create a storage layer across on-premises data centers, private clouds, and public cloud services such as Amazon S3, Azure Blob Storage, and Google Cloud Storage.

MinIO vs. Amazon S3

When you hear about MinIO as an object storage solution with full S3 API compatibility, your first question is probably, "How does it compare with S3?" Both are popular options

Key Features

One of the most prominent MinIO features is S3 application programming interface (API) compatibility, which makes it easy to integrate with existing applications and services built for Amazon S3. Since S3 was released in 2006 it has been heavily marketed and has been used by thousands of projects as one of the standard back-end options available. Whenever you use a popular solution that supports various storage back ends, chances are S3 is among them.

Similar to S3, MinIO is designed for high-speed, low-latency access to data, making it suitable for use in demanding applications and environments. Mind you, in this case, the overall performance depends on you – especially on the hardware components you choose. Amazon S3 is famous for its high availability (99.99%, or "four nines," for S3 Standard storage class), but this number is achieved by Amazon having multiple nodes in its data centers around the globe. Although Amazon doesn't publish official numbers, some experts estimate that at least four copies of each ob-

ject are stored in S3. If you need similar data durability and availability, you need to invest in hardware.

Another important feature is scalability: MinIO can scale seamlessly from a single node to a multinode distributed setup, allowing it to grow with your data storage needs. In terms of data protection, MinIO uses erasure coding and bit rot protection to ensure data durability and integrity. Additionally, it supports server-side and client-side encryption for data security.

Because MinIO is designed with a small footprint, it is suitable for deployment in containerized environments such as Docker and Kubernetes. It is also cost effective: By using commodity hardware and erasure coding for data protection, object storage can reduce the overall cost of ownership compared with traditional storage solutions. As with all object storage solutions, the flat address space and unique object identifiers make it easy to manage and access data, regardless of size.

for storing and managing vast amounts of unstructured data, but each comes with its own set of features and trade-offs.

Starting with similarities, the obvious one is the S3 API: Both MinIO and Amazon S3 use the same API for interacting with the storage service, making it easy to switch between them or integrate applications and services built for Amazon S3 with MinIO. They also employ the object storage model, which stores data as objects with unique identifiers and metadata in a flat address space. This structure enables seamless management and access to large amounts of unstructured data with excellent scalability: Both solutions offer virtually unlimited storage capacity, allowing you to scale horizontally across multiple nodes or clusters as your data storage requirements grow. Also, both MinIO and Amazon S3 provide data protection mechanisms, such as erasure coding and replication, to ensure the durability and availability of your data, even in the case of hardware failures.

As for differences, the first is related to deployment options. Amazon S3 is a managed cloud storage service provided by Amazon Web Services, whereas MinIO is an open source solution that can be deployed on-premises, in public clouds, or in hybrid cloud environments. This flexibility allows you to choose the deployment model that best meets your organization's requirements in terms of data sovereignty, compliance, and latency. In fact, if you wanted to, you could even deploy MinIO on AWS – although such a setup wouldn't make much sense in most cases.

Another practical difference lies in cost: Amazon S3 follows a

pay-as-you-go pricing model that is based on data storage, data transfer, and the number of requests. MinIO, on the other hand, is free to use and deploy, with costs associated only with the underlying infrastructure and optional commercial support subscriptions. MinIO, then, can be a more cost-effective option, particularly for organizations with large data storage needs or variable workloads, and is probably the main reason why MinIO is so popular in large organizations: When you work with petabytes of data, the cost of S3 becomes overwhelming, and at some point it makes sense to deploy a more cost-efficient solution. Because MinIO is an S3-compatible and battle-proven solution, it becomes the main choice in such scenarios.

Things become interesting when it comes to performance: MinIO is designed for high-speed, low-latency access to data and can deliver better performance than Amazon S3, especially in on-premises or private cloud deployments where network latency can be minimized. In other words, although you can't control the performance of S3, you can do a lot to optimize MinIO and make sure it works extremely fast if you eliminate bottlenecks.

As an open source solution, MinIO offers more customization and control over the storage system, allowing you to fine-tune configurations and even contribute to the project's development. Amazon S3, being a managed service, offers less control and customization, but also requires less management overhead.

As for security, both MinIO and Amazon S3 provide robust security

features, including encryption, access control, and audit logging. However, with MinIO, you have complete control over your data and security configurations, which can be an advantage for organizations with strict compliance and data privacy requirements.

Installation

Before you begin the installation process, ensure that your system meets the requirements listed in [Table 1](#).

In general, you have two ways to install and use MinIO: as a standalone server or as a multinode cluster. The first option is perfect for testing and initial evaluation; the second is used for production. Installing MinIO as a standalone server is a simple process, and the steps differ slightly depending on your operating system. In Linux, you start by downloading the MinIO server binary from the official MinIO website [\[1\]](#), make the downloaded binary executable, move the binary to the `/usr/local/bin` directory, and start the MinIO server with the default credentials and the console on port 9090:

```
wget https://dl.min.io/server/minio/2
  release/linux-amd64/minio
chmod +x minio
sudo mv minio /usr/local/bin
mkdir /data
minio server /data --console-address :9090
```

After a few moments you should see output similar to that in [Listing 1](#). At this point you should be able to connect to the console running on port 9090 ([Figure 1](#)) with the default credentials (*minioadmin* as the login and password). I show you how to change these insecure details later.

Although you could stop here, it is convenient also to install the auxiliary MinIO Client (*mc*). Users of the venerable Midnight Commander will note a name clash, so you will need to rename one program or the other. Start by downloading the client, making it executable, and moving it to somewhere in your `$PATH`:

Table 1: Before You Install

Factor	Requirement
Operating system	MinIO is compatible with Linux, macOS, and Windows operating systems. However, for production deployments, a Linux-based OS is recommended.
Hardware	MinIO requires a minimum 1GB of RAM and enough disk space for your data storage needs. For production environments, it's recommended to use multiple drives or nodes for data protection and performance. The faster the drives, the better the performance of the whole cluster.
Software	MinIO requires a modern web browser for its web-based management console, the MinIO Browser.

```
wget https://dl.min.io/client/mc/2
  release/linux-amd64/mc
chmod +x mc
sudo mv mc /usr/local/bin/mc
```

One of the most useful client commands to run just after the installation sets the so-called alias for your local deployment that will make it easier for you to refer to the deployment later. For example, the command

```
mc alias set mycluster1 2
http://127.0.0.1:9000 minio minio
```

names the local deployment *mycluster1* and sets the access key and secret key to *minio*.

Authorization and Authentication

MinIO supports several mechanisms for identity management, both internal and external. In the next example, I use the built-in identity management identity provider (IDP). In this case, you should create a new user, either in the console or with the *mc* utility.

For example, to add user *8pu2T6NBB6* for the deployment *mycluster1* defined earlier with the secret key *SPKKwJfTKl*, you would use the command:

```
mc admin user add mycluster1 2
8pu2T6NBB6 SPKKwJfTKl
```

To assign the built-in *readwrite* policy to the new user, enter:

```
mc admin policy set mycluster1 2
readwrite user=8pu2T6NBB6
```

Later, I will show you how to use familiar S3 bucket policies. Note that, by default, MinIO has a root user with its access and secret keys controlled by the environment variables:

```
MINIO_ROOT_USER
MINIO_ROOT_PASSWORD
```

If you decide to use them, make sure to use long random strings and rotate

them often. Nevertheless, the MinIO team discourages using the root user credentials and recommends creating users with reasonably limited access rights, as shown earlier.

Encryption

Ensuring data security is critical when deploying any storage solution, and MinIO provides built-in features for encrypting data both at rest and in transit. Encrypting data in transit ensures that your data is secure while being transferred between clients and the MinIO server. The most common method to achieve this is by using Transport Layer Security (TLS) encryption.

To enable TLS for your MinIO deployment, you need to obtain a valid TLS certificate from a trusted certificate authority (CA). Alternatively, you can generate a self-signed certificate for testing purposes with tools such as OpenSSL or Let's Encrypt. Keep in mind that self-signed certificates might not be suitable for production environments because of potential trust issues.

To configure MinIO with your TLS certificate and private key, you need to place both files (i.e., *cert.pem* and *private.key*) on the MinIO server. Next, set the following environment variables when starting the MinIO server:

```
MINIO_SERVER_CERT_FILE: /<path to>/cert.pem
MINIO_SERVER_KEY_FILE: 2
/ <path to>/private.key
```

Listing 1: server Command Output

```
API: http://192.0.2.10:9000 http://127.0.0.1:9000
RootUser: minioadmin
RootPass: minioadmin

Console: http://192.0.2.10:9090 http://127.0.0.1:9090
RootUser: minioadmin
RootPass: minioadmin

Command-line: https://min.io/docs/minio/linux/reference/minio-mc.html
$ mc alias set myminio http://192.0.2.10:9000 minioadmin minioadmin

Documentation: https://min.io/docs/minio/linux/index.html

WARNING: Detected default credentials 'minioadmin:minioadmin', we recommend that you change these
values with 'MINIO_ROOT_USER' and 'MINIO_ROOT_PASSWORD' environment variables.
```

Make sure to replace *<path to>* with the correct paths to your certificate and private key files.

Encryption at rest is equally important: It protects your data from unauthorized access while it is stored on disk. MinIO supports three server-side encryption (SSE) methods:

- server-side encryption with per-bucket keys (SSE-KMS),
- server-side encryption with per-deployment keys (SSE-S3), and
- server-side encryption with client-managed keys (SSE-C).

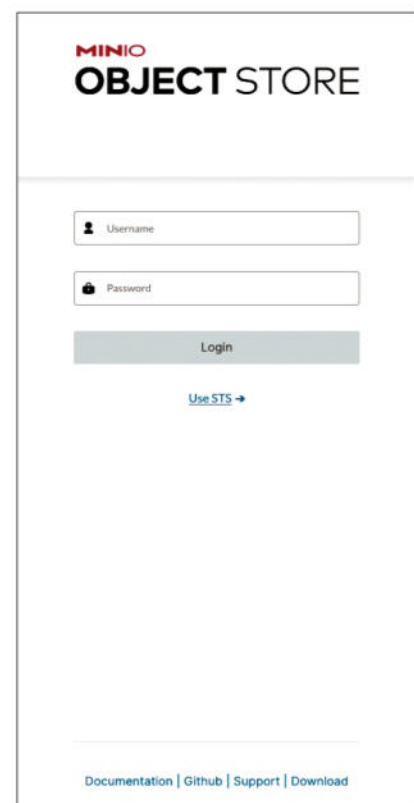


Figure 1: The MinIO console runs on port 9090.

You can find comprehensive documentation about these methods on the MinIO website [2].

Roughly, you have two basic approaches to managing encryption keys: They can be stored externally by a back-end service such as Vault, or they can be managed by the client. The following key management service (KMS) back ends are currently supported:

- AWS Secrets Manager
- Google Cloud Secret Manager
- Azure Key Vault
- HashiCorp Vault

The last option can be installed on-premises or on any public or private cloud and plays nicely with MinIO. If you want to use externally managed keys, you need to install the Key Encryption Service (KES) [3] utility first. Make sure you read the relevant documentation thoroughly because you can inadvertently lose access to your data if you make a mistake at this point.

General Security Considerations

Keeping your MinIO server and client up to date is essential for maintaining security and benefiting from the latest features and improvements. Regularly check for updates and apply them as needed. To update the server, use the command:

```
mc admin update mycluster1
```

In addition to securing your MinIO deployment, it's crucial to implement network security measures to protect your infrastructure. As a rule, MinIO should be deployed behind a firewall, and access should be limited to specific IP addresses or networks. You can also implement an intrusion detection and prevention system (IDPS) to monitor network activity and identify potential security threats.

Performance

Once MinIO is installed, you might want to configure it for better performance according to your specific use

case. Here are some configuration options to consider:

Erasure Coding. By default, MinIO uses erasure coding for data protection. You can adjust the number of data and parity drives to balance performance and redundancy. To do this, modify the `minio server` command by specifying the number of data and parity drives:

```
minio server --data 6 --parity 2 /data
```

Readjust the erasure code setup according to your desired balance between performance, storage efficiency, and data protection. For example, as far as performance goes, increasing the number of data drives generally improves the read and write performance because the data is distributed across more drives, which can be accessed in parallel. However, adding more parity drives can increase the overhead of encoding and decoding, which could affect performance.

As for storage efficiency, the more parity drives you use, the more storage space is consumed by redundant information, reducing overall storage efficiency because less space is available for data. However, specifying fewer parity drives can increase the risk of data loss or corruption because of less redundancy in the system. A similar balance needs to be achieved for data protection: Increasing the number of parity drives enhances data protection because it allows the system to recover from a higher number of simultaneous drive failures or data corruption. However, adding more parity drives comes at the cost of reduced storage efficiency and potentially lower performance. For bit rot protection, MinIO uses HighwayHash by default, which provides a good balance between performance and security. However, you can choose a different hash algorithm if you have specific performance or security requirements.

For larger deployments, you should definitely use MinIO in a distributed mode to scale across multiple nodes. In very rough terms, to set up a

distributed cluster, specify the drives or directories on each node when starting the MinIO server:

```
minio server 2
http://node1.example.com/data 2
http://node2.example.com/data 2
...
```

At the same time, use a load balancer (e.g., HAProxy or NGINX) to distribute client requests evenly across the MinIO instances. This step can help prevent bottlenecks and ensure high availability.

Setting up a MinIO production system deserves a whole book and cannot be covered in a short article. Fortunately, the documentation on the project website is complete and frequently updated. You will probably want to use Terraform and Ansible to keep your infrastructure and software configuration in code, preferably in a repository.

Selecting the appropriate hardware for your MinIO deployment is crucial for achieving optimal performance. MinIO is designed to work with modern hard and solid-state drives (HDDs and SSDs), so choose drives with high input/output operations per second (IOPS) and low latency for best performance. MinIO can also benefit from multicore processors because it parallelizes operations across multiple cores; therefore, opt for a processor with a high core count and clock speed. Similarly, MinIO's performance is heavily dependent on network speed. Choose high-bandwidth network interfaces (10Gbps or higher) to minimize network bottleneck issues. For the filesystem, MinIO recommends XFS or ext4 for optimal performance. Both filesystems are stable, widely used, and have proven performance in large-scale deployments. If you need to maximize performance, you can achieve it by several popular methods, such as disabling access time updates: When mounting the filesystem, add `noatime` to the mount options:

```
/dev/sdb1 /mnt/data ext4 2
defaults,noatime 0 0
```


You can also increase the read-ahead value to boost the sequential read performance with the `blockdev` command. For example, to set a read-ahead value of 2048 (1MB) for the `/dev/sdb1` device, use:

```
blockdev --setra 2048 /dev/sdb1
```

For kernel parameters, you can consider increasing the maximum number of open file descriptors because MinIO can use a large number of these during operation, especially in large-scale deployments. In this case, increase the `fs.file-max` kernel parameter value to allow for more open file descriptors. You can set this value temporarily with the command:

```
sysctl -w fs.file-max=1000000
```

To make this change permanent, add the following line to `/etc/sysctl.conf`:

```
fs.file-max = 1000000
```

In a similar way, you can increase the maximum socket buffer size to enhance network performance with the `net.core.rmem_max` and `net.core.wmem_max` kernel parameters. You can set these values temporarily with the commands:

```
sysctl -w net.core.rmem_max=4194304
sysctl -w net.core.wmem_max=4194304
```

As before, to make these changes permanent, add the following lines to `/etc/sysctl.conf`:

```
net.core.rmem_max = 4194304
net.core.wmem_max = 4194304
```

Another popular approach is to configure the TCP keepalive settings to improve the stability of long-lived connections by setting some kernel parameters. You can set these values temporarily with the commands:

```
sysctl -w net.ipv4.tcp_keepalive_time=120
sysctl -w net.ipv4.tcp_keepalive_intvl=30
sysctl -w net.ipv4.tcp_keepalive_probes=3
```

As before, you can add them to `/etc/sysctl.conf` to make the changes

permanent. When setting these kernel options, don't stick to the numbers provided above but read the documentation and experiment with your own values to find the right balance.

If the `tcp_keepalive_time` and `tcp_keepalive_intvl` values are set too high, it might take longer for the system to detect idle or dead connections that could result in increased latency and reduced performance, especially when resources are tied up in maintaining these idle connections. On the other hand, if the `tcp_keepalive_time` and `tcp_keepalive_intvl` values are set too low, the system will send keepalive probes too frequently, which can lead to unnecessary resource usage, both in terms of CPU and network bandwidth, as well as potential congestion on the network.

As a side note, you need to be prepared for sporadic problems with faulty hardware. Specifically, when one of the drives in a cluster starts to fail, your safest bet is to remove it. Because MinIO tries to write all the data to the faulty drive, if the operations start to be painfully slow, it can affect the whole storage back end, so removing the faulty drive should help. Fortunately, this is not a very common problem.

The MinIO Client (`mc`) plays a crucial role in interacting with your MinIO server. To optimize its performance, first use the `--concurrent` flag to increase the number of concurrent operations when mirroring, copying, or removing objects. This step can help speed up these tasks, especially when dealing with large numbers of small files. Second, enable client-side compression with the `--compress` flag when uploading objects to save bandwidth and reduce transfer times. Moreover, MinIO supports caching of frequently accessed objects, which can significantly reduce latency and improve performance for read-heavy workloads. When starting the MinIO server, configure the cache subsystem by specifying the `MINIO_CACHE` environment variable:

```
export MINIO_CACHE_DRIVES=2
"/mnt/cache1",/mnt/cache2"
minio server /data
```

Replace `</mnt/cache1>`, `</mnt/cache2>` with the paths to your cache drives. Finally, the way you organize your objects in MinIO buckets can have a significant affect on performance. To distribute objects evenly across the namespace and help prevent performance issues caused by having too many objects in a single directory, use a hierarchical structure with multiple levels of directories. At the same time, avoid using overly deep directory structures because they can increase latency for object retrieval operations.

Buckets and Objects

To create a new bucket, use the `mb` subcommand followed by the alias and bucket name:

```
mc mb mycluster1/new-bucket
```

To list all the buckets in your MinIO server, use the `ls` subcommand:

```
mc ls mycluster1
```

In the same way, to upload a file to a bucket and download a file from a bucket, use the `cp` subcommand,

```
mc cp local-file.txt mycluster1/new-bucket
mc cp 2
mycluster1/new-bucket/local-file.txt 2
local-file.txt
```

just reversing the order of its arguments. Removing an object or bucket is equally simple:

```
mc rm mycluster1/new-bucket/local-file.txt
mc rb mycluster1/new-bucket
```

The same actions can be performed in the browser ([Figure 2](#)), although the `mc` utility is probably more convenient and less prone to error, especially for larger collections of data.

Bucket Policies

Granular access control is essential for managing user access to your data. MinIO supports bucket policies and user policies, which allow

you to control access to specific resources on the basis of users, groups, and actions. AWS users will find MinIO bucket policies familiar (**Listing 2**).

To use the policy from **Listing 2**, you need to save it first (e.g., as `read-only-policy.json`) and then register it (e.g., as `read-only-policy`):

```
mc admin policy add mycluster1 ?
read-only-policy read-only-policy.json
```

At this point you can assign it with

Listing 2: Sample Bucket Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::my-minio-bucket/*"
      ]
    }
  ]
}
```

```
mc admin policy set mycluster1 ?
read-only-policy user=new-user
```

just like the built-in policy used earlier.

Object Versioning and Lifecycle Policies

Object versioning and lifecycle policies provide additional layers of data protection and help manage storage costs. Enabling object versioning to maintain multiple versions of objects in your MinIO buckets allows you to recover previous versions in case of accidental deletion or modification. To enable object versioning, use the MinIO Client as follows:

```
mc versioning mycluster1/my-minio-bucket ?
enable
```

To create and apply lifecycle policies, use `mc ilm`; for example:

```
mc ilm add mycluster1/my-minio-bucket ?
--id "ExpireOldVersions" ?
--status "Enabled" ?
--expiry-days 30
```

With this command, you can automate the management of object versions, such as transitioning older versions to a different storage class or deleting them after a specified period.

Migration from S3

Organizations might consider migrating from S3 to MinIO for three main reasons: cost savings, improved performance, and data sovereignty requirements. Although both services are S3 compatible, migrating data between them requires careful planning and execution. The following steps are by no means exhaustive; rather, they present a basic outline to help you migrate your data from Amazon S3 to MinIO.

The first step is to install MinIO and define aliases in `mc` – both for MinIO and S3:

```
mc alias set myminio ?
http://<your-minio-server>:9000 ?
<minio-access-key> <minio-secret-key>
mc alias set aws ?
https://s3.amazonaws.com ?
<aws-access-key> <aws-secret-key>
```

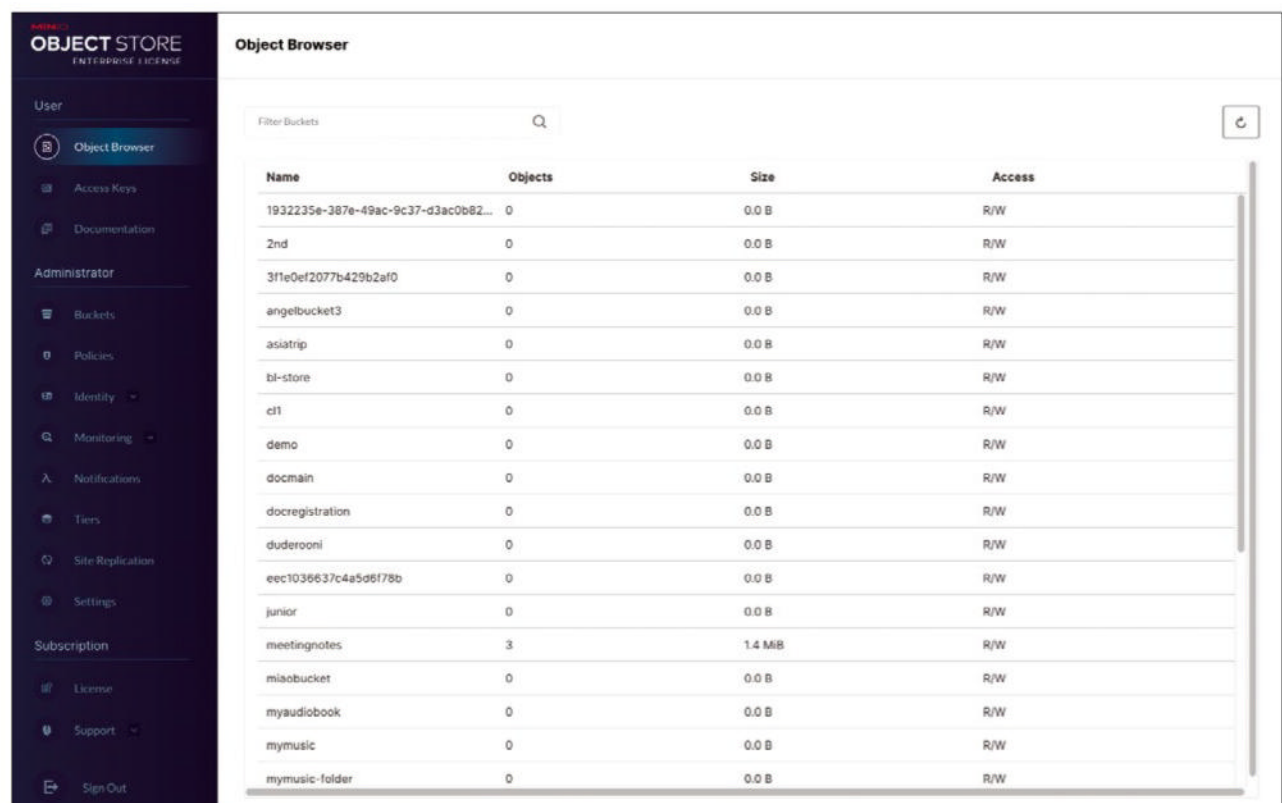


Figure 2: The MinIO browser.

Be sure to replace the placeholders (<>) with the appropriate access keys, secret keys, and server addresses. Next, use the `mc mirror` command to transfer data from your Amazon S3 buckets to your MinIO server. This command synchronizes the source and destination, copying objects that don't exist in the destination or have a different size or modification time:

```
mc mirror ?
  --overwrite ?
  --remove aws/your-s3-bucket ?
  myminio/your-minio-bucket
```

The `--overwrite` flag tells `mc` to overwrite the destination objects if they differ from the source objects, whereas the `--remove` flag allows `mc` to delete destination objects that no longer exist in the source. Note that transferring large amounts of data can take a significant amount of time depending on your network and storage conditions. Consider running the `mc mirror` command during periods of low system usage or scheduling it to run during off-peak hours. Also, mind your AWS bill – egress fees are notoriously high, and if you have particularly large volumes of data that are not particularly important, it might make more sense just to delete them.

After migrating your data, update your applications to use the MinIO server instead of Amazon S3. This step might involve changing the S3 endpoint, access keys, and secret keys in your application's configuration. Because both Amazon S3 and MinIO use the S3 API, no major code changes should be necessary. Once your data has been migrated and your applications have been updated, it's essential to verify data integrity, which you can do by:

- comparing the object counts and total size of your Amazon S3 buckets and MinIO buckets,
- performing test queries and operations on the data in your MinIO server to ensure that it behaves as expected, and

- validating the migrated data with application-specific tests, such as searching for particular records, generating reports, or processing data.

After verifying that your data has been successfully migrated to MinIO and that your applications are functioning correctly, you can decommission your Amazon S3 buckets. Be sure to delete any unnecessary objects, versioned objects, and bucket configurations before deleting the bucket itself.

In real-life scenarios, migration is rarely that simple, though. You might have a huge number of buckets with objects in them, each with a different access configuration by AWS Identity and Access Management (IAM) user, group, and role policies, as well as individual bucket policies, not to mention organizational-level policies (e.g., service control policies (SCPs)). Replicating all this complexity might be difficult, even if you use IAM Roles Anywhere. If you don't have proper tag policies in place, you could even have a hard time figuring out who is the owner of a given bucket, and contacting them might not be easy. When you take all this and other aspects into consideration, the process of copying data itself might be the easiest part.

Logging and Monitoring

Monitoring and auditing MinIO server activity can help identify security threats and ensure compliance with data protection policies. MinIO supports logging of audit events and integration with monitoring tools like Prometheus for real-time insights.

To enable audit logging, add the `--audit` flag when starting the MinIO server:

```
minio server --audit /data
```

Also, consider monitoring MinIO performance with built-in Prometheus

metrics, which provide insights into system health, resource usage, and performance metrics. To enable these metrics, add the `--metrics prometheus` flag when starting the MinIO server:

```
minio server --metrics prometheus /data
```

You can easily integrate it with tools such as Grafana to create dashboards and alerts for system performance and potential security issues.

Conclusion

MinIO is a powerful, open source object storage solution that offers a range of features and benefits for organizations looking to store and manage large amounts of unstructured data. MinIO's durability and redundancy features make it an excellent choice for backup and disaster recovery solutions. By storing backups as objects in MinIO, organizations can ensure data protection and quick recovery in case of hardware failures or other disasters. MinIO is also well-suited for media storage and streaming applications, such as video-on-demand platforms and content delivery networks. Its high-performance and scalable architecture can handle the storage and delivery of large media files with low latency.

Still, Amazon S3 is a great alternative if your workloads are already on AWS, you don't work with large datasets, or you would prefer not to deal with managing MinIO yourself. You need to remember that to achieve S3 availability, you will need to replicate data across multiple MinIO clusters around the world, so a careful cost-benefit analysis should come first. ■

Info

[1] MinIO packages: [\[https://min.io/download\]](https://min.io/download)

[2] SSE-KMS docs: [\[https://min.io/docs/minio/linux/administration/server-side-encryption/server-side-encryption-sse-kms.html\]](https://min.io/docs/minio/linux/administration/server-side-encryption/server-side-encryption-sse-kms.html)

[3] Key Encryption Service: [\[https://github.com/minio/kes\]](https://github.com/minio/kes)

Six new security features
for Windows Server 2022

Shielded

Configure the Secured-core server components to reduce the attack surface of your system with minimal overhead. By Christian Knermann

At first glance, Windows Server 2022 looks like a carefully modeled update of its direct ancestor, Windows Server 2019. Microsoft has clearly focused on evolution instead of revolution. However, a second glance reveals exciting new features, especially in the field of information security. Secured-core server sees Microsoft add no fewer than six components to its new operating system that boost the system's security with minimal overhead. Whether an instance of Windows Server 2022 supports these features, and which ones, depends on the underlying hardware and whether you have a physical or virtual machine; the firmware and, in the case of a VM, the hypervisor, also need to be compatible.

Differences

Before I go into the details of and technical requirements for Secured-core server, I would like to prevent a misunderstanding. Microsoft also used the term “core” in connection with earlier editions of Windows Server, but that meant an installation without a graphical user interface. As early as Windows Server 2008 R2, you were allowed to choose between a core installation and a full

installation with desktop display as part of the setup.

A Secured-core server and its functionality are different. You can use all of the new security features in conjunction with a graphical user interface in all editions. Windows Server 2022 Standard, Datacenter, and Datacenter Azure Edition support Secured-core servers, provided the hardware and virtualization are ready for them.

Secure Boot Explained

Security starts as early as the BIOS, or more likely in a state-of-the-art Unified Extensible Firmware Interface (UEFI) that supports the Secure Boot standard, which is not a Microsoft invention but a part of the UEFI specification determined by various original equipment manufacturers (OEMs). Secure Boot starts even before an operating system fires up and is intended to verify the firmware's integrity and to lock out rootkits. To do this, the UEFI checks the signatures of its boot code and firmware drivers. If this check fails, the firmware triggers a process defined by the OEM to restore a trusted state [1]. Similarly, the firmware also verifies the operating system's boot manager. The firmware

will only hand over control for further startup if it also has a valid signature. The other components of the operating system, such as the kernel and device drivers, must also prove by means of signatures that they have not been changed. If any of the signatures do not match those in the UEFI database, the system will not boot. Under the hood, Secure Boot uses asymmetric encryption, much like a public key infrastructure (PKI). Like a certification authority (CA), UEFI firmware manufacturers form the root of trust. They own the platform keys (PKs) with which they demonstrate the authenticity and genuineness of their firmware. Manufacturers such as Microsoft sign their operating systems and drivers with key exchange keys (KEKs), which an OEM stores in its firmware at the factory before blocking any write access. Subsequent updates of the KEK database are possible but need to be signed with the PK. Microsoft has a KEK that it can use to sign new versions of Windows and other software components and drivers or to block existing signatures. This procedure is known as the static root of trust for measurement (SRTM).

TPM 2.0

As a further root of trust, the Secured-core server uses an active Trusted Platform Module (TPM) version 2.0. The TPM chip generates, stores, and

controls access to cryptographic keys and hash values. Microsoft uses a TPM for biometric logins with Windows Hello on client computers and for BitLocker drive encryption.

As part of a measured boot, the TPM generates and stores hashes for all components involved in the system startup process [2]. However, these hashes cannot be verified by a trusted server with just on-board tools (remote attestation); instead, this requires Intune, the Microsoft Endpoint Configuration Manager, or some third-party software. However, even without additional applications and services, a TPM helps enhance security because it forms the basis for another component of the Secured-core server: system monitoring.

Microsoft Defender System Guard

System Guard is a component of Microsoft Defender and follows the “assume breach” principle; that is, a system and its components are assumed to be fundamentally untrustworthy and potentially already compromised. Accordingly, system monitoring does not rely on Secure Boot having proven the integrity of the UEFI beyond doubt but, instead, it relies on the additional Secure Launch procedure and on safeguarding the CPU’s System Management Mode (SMM) [3].

Microsoft also refers to Secure Launch as a dynamic root of trust for measurement (DRTM), because unlike Secure Boot, this procedure does not rely on static lists of trusted and revoked signatures. Instead, Defender System Guard uses the TPM itself to compute checksums for the firmware, the rest of the hardware configuration, and the operating system components, which it compares with previous states to detect anomalies. Additionally, system monitoring secures the SMM of the processor. Because code in this mode runs with the highest privileges, system monitoring uses security features implemented in the hardware of a state-of-the-art processor and establishes protections

for access to sensitive areas of main memory.

Protection Against Peripherals

Boot DMA protection builds on kernel DMA protection that Microsoft first introduced in Windows 10 version 1803 [4]. This security feature protects the system from drive-by attacks with hot-pluggable peripherals. This attack vector aims to read sensitive data from main memory by direct memory access (DMA) or to inject malware directly into a system past the lock screen.

Boot DMA protection protects the external and internal PCI and PCIe interfaces of a system against these attempts during the boot phase and at runtime. However, this protection requires the device drivers of the external devices to support DMA remapping.

Security Through Virtualization

Virtualization-based security (VBS) uses features of Microsoft’s Hyper-V hypervisor to store sensitive information, such as password hashes, in a specially secured memory area. Even privileged system processes cannot easily access this isolated memory area. VBS requires UEFI version 2.6 or later with support for a memory attributes table (MAT), which ensures a clean segregation of the runtime memory areas of code and data. VBS forms the basis for the sixth and final component of Secured-core server: hypervisor enforced code integrity (HVCI). This function watches over the execution of code in kernel mode and only permits execution if the code can be verified as legitimate, including device drivers, any software that uses the Control Flow Guard (CFG) function, and certificates. If you want to install Windows Server 2022 directly on physical hardware, the best case scenario is that you can use all six features of Secured-core server – assuming the hardware, firmware, and device drivers support it.

Checking Hardware Requirements

If you start with a fresh installation of a physical system that has not yet run a Secured-core server, you first need to make sure that your hardware has and uses a modern UEFI. Current systems that are certified to run Windows Server 2022 come with the necessary prerequisites. The chances are still good if you are using hardware that is three to five years old. If the option of a legacy BIOS mode does exist in the firmware settings for downward compatibility with older operating systems, it must be disabled. The further steps for enabling hardware-supported virtualization differ depending on the processor manufacturer, the other server hardware, and the UEFI version.

Hardware virtualization is called Intel VT-d/VT-x or AMD IOMMU, depending on the CPU vendor, but it can also be hidden under other terms or in a submenu depending on the manufacturer and version of the UEFI. Look for menu items such as *Advanced*, *Processor Configuration*, *CPU Configuration*, *System Configuration*, *Chipset*, *Security*, or *Northbridge*. Intel simply refers to hardware virtualization as Intel Virtualization Technology, VT-x, or on older systems, perhaps Vanderpool after the original internal code name from the development of this technology. AMD also refers to the whole thing as the Secure Virtual Machine (SVM) or AMD-V.

The Secure Boot and TPM configuration can be found in areas with titles like *Advanced*, *Security*, or *Trusted Computing*. Alternative locations for configuration are also *Security Device*, *Security Device Support*, *AMD fTPM*, *AMD PSP fTPM*, or *Intel Platform Trust Technology* (PTT). If in doubt, the documentation of the server or mainboard manufacturer will help.

Configuring Virtual Machines

System monitoring and DMA protection at startup time are reserved for instances of Windows Server 2022 that you install directly on a physical system. You can also use the

remaining four components on virtual machines if the underlying hypervisor passes the necessary functions to VMs. To do this, create a second-generation VM in Microsoft Hyper-V and check the *Enable Secure Boot* and *Enable Trusted Platform Module* options in the *Security* area. Hyper-V meets the requirements for VBS and HVCI without any further action. For virtualization with VMware vSphere or ESXi, you need to enable VBS when creating a new VM in the sixth dialog step that determines the guest operating system family and version by checking the *Enable Windows Virtualization Based Security* [5] box. In the dialog that follows, define your VM's hardware configuration. You can add a virtual TPM by selecting *Add New Device*, but this only works if you have configured a key provider in your vSphere environment up front [6].

Securing VMs in Azure and Azure HCI

What works for VMs in your on-premises data center works just as well in the cloud. Microsoft supports the Secured-core server components in both its Azure Cloud and Azure Stack hyperconverged infrastructure (HCI). As part of this infrastructure, Microsoft,

in cooperation with OEMs, offers certified hardware that you can run in your own data center while still managing it in a unified way with your resources in the cloud in the Azure portal. If you want to install a Secured-core server in Azure, it is important to note that you need to make this decision when you create the VM. Secured-core server cannot be retrofitted to existing VMs, leaving a new installation as your only option. You will find the settings for this on the first page of the New Virtual Machine Wizard. In the *Security type* drop-down box, select *Trusted launch virtual machines* instead of *Standard*. As on a local Hyper-V host, this is a second-generation VM [7]. A link will then appear below the drop-down box; you can use it to configure the VM's security features. Two options, *Secure boot* and *vTPM*, are already enabled by default.

Now you have all the prerequisites for the Secured-core server in place at the hardware and virtualization levels. Everything else is done at the operating system level.

Windows Admin Center Control

Microsoft has been promoting the web-based Windows Admin Center

(WAC) as a modern alternative to the outdated Server Manager for quite some time. If you have not used WAC before, now is a good time to do so. WAC gives you an overview of the status of all the Secured-core server components, and you can enable functions that have not yet been configured (Figure 1).

You can download the latest version [8] for free and install it on a Windows server; it does not need to be one of the machines you want to configure as a Secured-core server. In production, you need to issue an SSL certificate for the fully qualified DNS name of your WAC server with an Active Directory (AD)-integrated or external PKI before the install. The setup is just a few steps. WAC sends diagnostic data to Microsoft. You can only choose between the mandatory data or additional optional data, and you can decide whether or not WAC should use the Microsoft update service for updates. The wizard defaults to TCP port 443 for the web front end. At this point you can also specify the fingerprint of your own certificate, if available. Alternatively, the setup generates a self-signed certificate that is only valid for 60 days. The wizard guides you through the last step of dialog. Click on the URL, which will look like <https://>

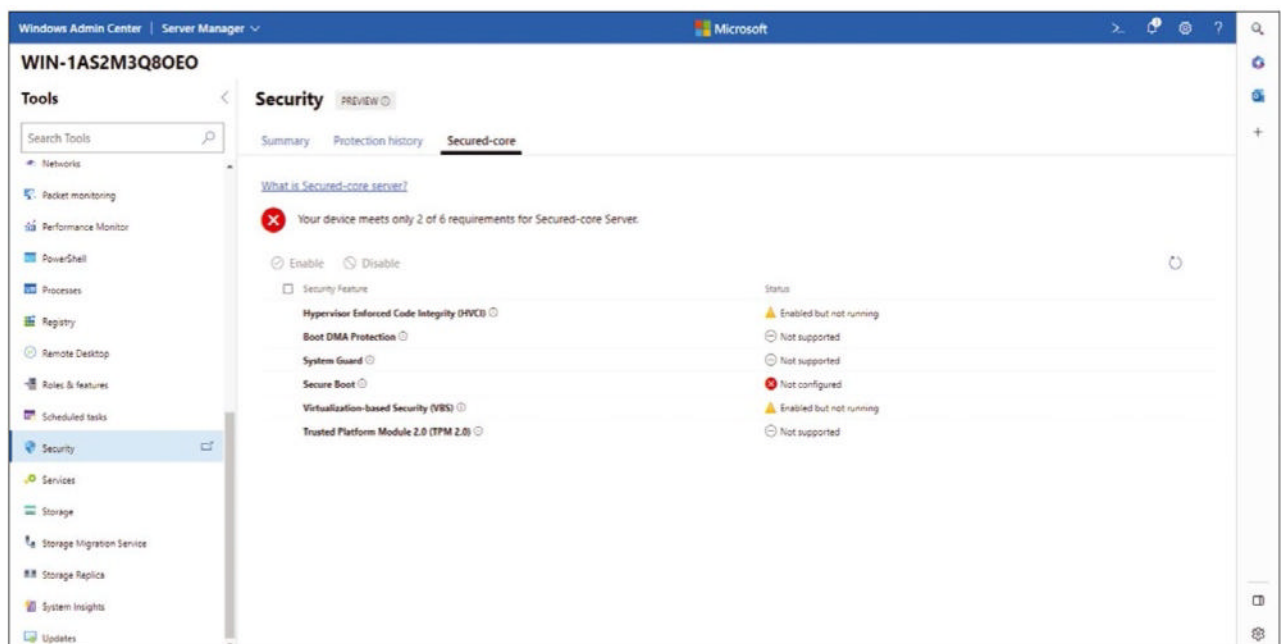


Figure 1: The Windows Admin Center provides information about the status of the Secured-core server functions.

< fully qualified domain name of server > : < port > or transfer it to your browser. The WAC then prompts you to log in. The server on which the WAC is installed acts as a gateway for connecting to and managing other systems. With the *All Connections* button in the window header, you can then add additional servers, client computers, and clusters. Navigate to the *All connections | Server Manager* area and use the + *Add* button to set up connections to your instances of Windows Server 2022.

If you now click on one of the server connections, WAC prompts you to authenticate again. If you want to avoid this in the future, you need to set up Kerberos restricted delegation from WAC to the respective target server, which can be done in PowerShell on a domain controller or a server with the AD snap-ins [9] installed. In the detailed view of the respective server, select the *Security* item from the

vertical navigation bar. In the main area of the page, you can then access the *Secured-core* tab to view the status of the six security features.

TPM 2.0 and Secure Boot

You cannot influence Secure Boot and TPM at this point. If WAC indicates *Not supported* for the two functions, your system either does not have the appropriate hardware or the functions are simply not enabled in the UEFI or the properties of the VM. As soon as you enable Secure Boot and TPM in the firmware of a physical system or the properties of a VM, the status in WAC changes to *On*. The two functions are automatically enabled with no further configuration options.

Enabling Functions with WAC

Assuming your machine meets the requirements, WAC displays a status

of *Not configured* for the other functions. In each case, check the box in the first column and click the *Enable* button to change the status of the components to *Enabled but not running*; the WAC then requests a server restart, which you can trigger immediately in the interface or schedule for a suitable time. The status of the functions then changes to *On*. Virtual instances of Windows Server 2022 have to do without DMA protection at boot time, as well as system monitoring; they have reached their highest protection level with four out of six Secured-core functions. Physical machines can also enable the remaining two options if all of the device drivers are compliant.

Fine Tuning with Group Policies

As an alternative to WAC, you can control the functions of the

Shop the Shop

sparkhaus-shop.com

Missed an issue?

You're in luck.

Most back issues are still available. Order now before they're gone!

shop.linuxnewmedia.com



GET IT NOW!
SAVE TIME ON
DELIVERY WITH OUR
ALTERNATIVE
PDF EDITIONS



Secured-core server through group policies. The options are found in the Group Policy Management Editor under *Computer Configuration | Policies | Administrative Templates | System | Device Guard*. Use the *Turn on Virtualization Based Security* setting when you get there.

If you enable this setting, four drop-down boxes in the options area control the details (Figure 2). The first option for the platform security level enables Secure Boot with or without DMA protection. *Not Configured* means that the group policy will not change the HVCI status already in place on a target machine. *Enabled without lock* is equivalent to the state that you can enable in WAC. HVCI is enabled in this case and can also be disabled again with a group policy. The *Enabled with UEFI lock* option links the HVCI status to the target's local UEFI, if compatible. In this case, the function can no longer be disabled remotely, whether by WAC or by group policy.

The memory attributes table checkbox ensures that VBS and HVCI are only used on systems with a compatible UEFI. Microsoft otherwise warns about crashes, data loss, or incompatibility when you install physical expansion cards. Credential Guard is not one of the six core functions of the Secured-core server, but it uses VBS as a basis for protecting credentials. The options are identical to those of HVCI. Credential Guard can also be connected to a local UEFI. Finally, the Secure Boot configuration option controls whether or not system monitoring should be enabled. In *Computer Configuration | Policies | Administrative Templates | System | Kernel DMA Protection* you find *Enumeration policy for external devices incompatible*

with *Kernel DMA Protection*. This setting decides whether the target system generally blocks or allows devices whose drivers do not support DMA remapping. By default, the third option specifies that these devices only work as long as a user is logged in and the screen is not locked.

Conclusions

Secured-core server uses all the security features of modern hardware and virtualization infrastructures. Provided UEFI and device drivers are suitable, the setup involves just a few steps. Secured-core server reduces the attack surface of Windows Server 2022 at no additional cost and with the least possible overhead. ■

Info

- [1] OEM process for Secure Boot: [\[https://learn.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-secure-boot\]](https://learn.microsoft.com/en-us/windows-hardware/design/device-experiences/oem-secure-boot)
- [2] Using TPM on Windows: [\[https://learn.microsoft.com/en-us/windows/security/information-protection/tpm/how-windows-uses-the-tpm#tpm-in-windows\]](https://learn.microsoft.com/en-us/windows/security/information-protection/tpm/how-windows-uses-the-tpm#tpm-in-windows)
- [3] Root of trust for protecting Windows: [\https://learn.microsoft.com/en-us/windows/security/threat-protection/windows-defender-system-guard/

[how-hardware-based-root-of-trust-helps-protect-windows\]](#)

- [4] Kernel DMA protection for Thunderbolt: [\[https://learn.microsoft.com/en-us/windows/security/information-protection/kernel-dma-protection-for-thunderbolt\]](https://learn.microsoft.com/en-us/windows/security/information-protection/kernel-dma-protection-for-thunderbolt)
- [5] Virtualization-based security for Windows: [\[https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.security.doc/GUID-CE292D3F-D4AC-4607-B262-DE19CE6E9F6B.html\]](https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.security.doc/GUID-CE292D3F-D4AC-4607-B262-DE19CE6E9F6B.html)
- [6] Configuring a key provider: [\[https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.security.doc/GUID-3D39CBA6-E5B2-43E2-A596-B9A69B094558.html\]](https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.security.doc/GUID-3D39CBA6-E5B2-43E2-A596-B9A69B094558.html)
- [7] Trusted launch: [\[https://learn.microsoft.com/en-us/azure/virtual-machines/trusted-launch\]](https://learn.microsoft.com/en-us/azure/virtual-machines/trusted-launch)
- [8] WAC download: [\[https://www.microsoft.com/en-us/windows-server/windows-admin-center\]](https://www.microsoft.com/en-us/windows-server/windows-admin-center)
- [9] Configuring single sign-on: [\[https://learn.microsoft.com/en-us/windows-server/manage/windows-admin-center/configure/user-access-control#configure-single-sign-on\]](https://learn.microsoft.com/en-us/windows-server/manage/windows-admin-center/configure/user-access-control#configure-single-sign-on)

Author

Christian Knerrmann is Head of IT-Management at Fraunhofer UMSICHT, a German research institute. He's written freelance about computing technology since 2006.

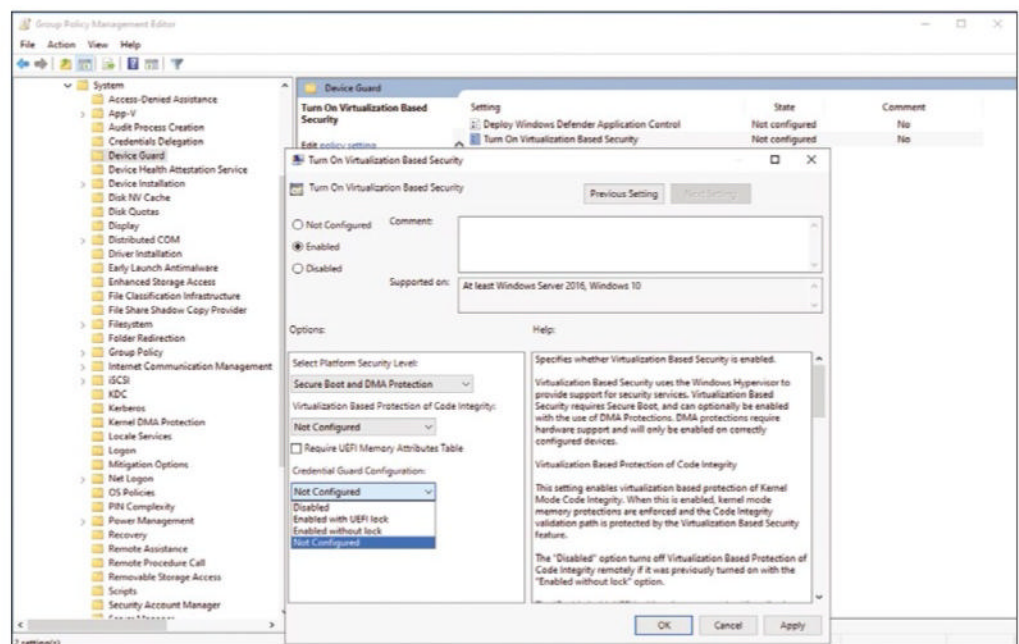
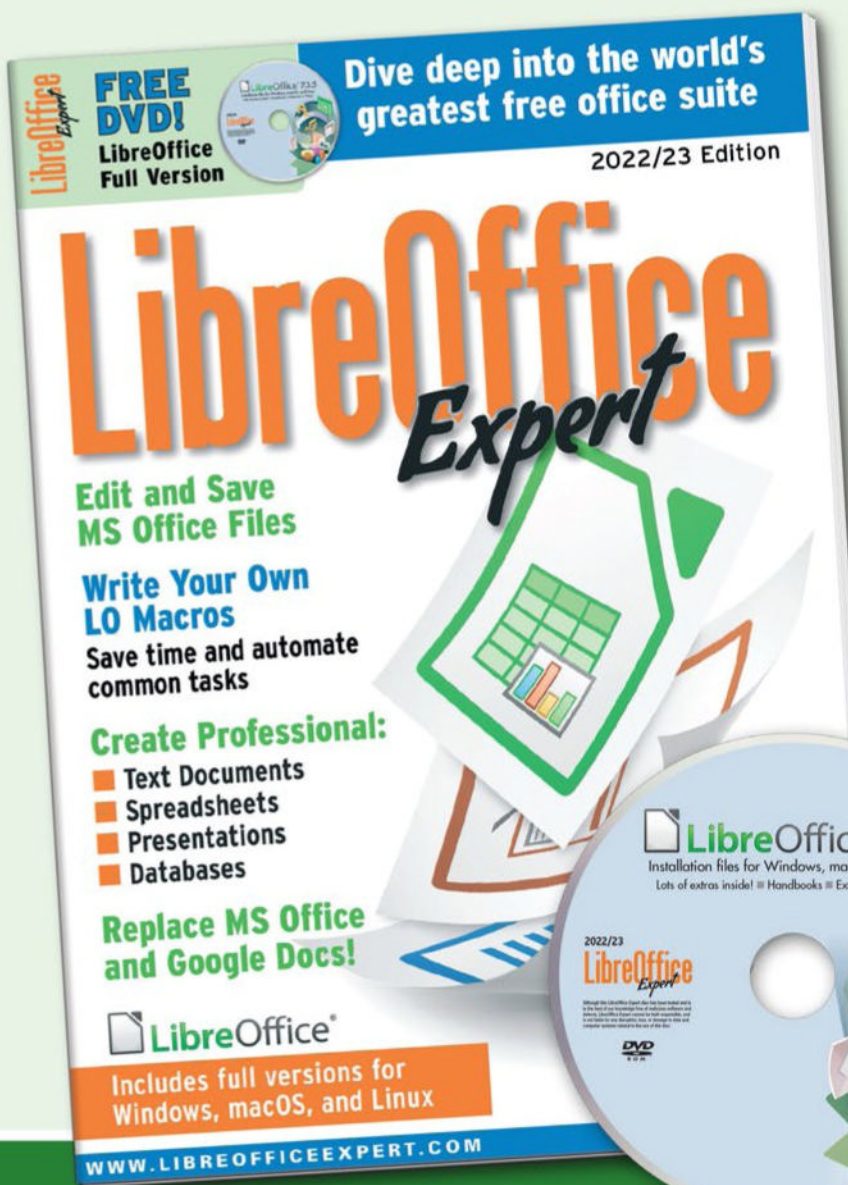


Figure 2: Group policies centrally control the functions of the Secured-core server.

Shop the Shop
sparkhaus-shop.com

Become a LibreOffice Expert

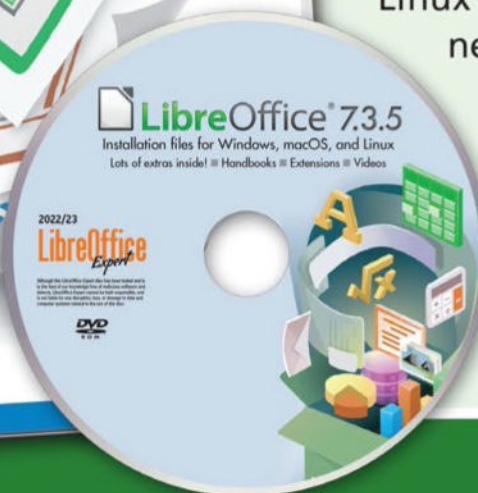


Explore the **FREE** office suite used by busy professionals around the world!

Create Professional:

- Text Documents
- Spreadsheets
- Presentations
- Databases

Whether you work on a Windows PC, a Mac, or a Linux system, you have all you need to get started with LibreOffice today. This single-volume special edition will serve as your guide!



Order online:
sparkhaus-shop.com/specials

For Windows, macOS, and Linux users!

Test mechanisms for best practices in cloud design

Best Clouds

Develop resilient and efficient cloud infrastructures for enterprise applications with the AWS Well-Architected Framework. We show you how to implement the solutions from the framework in practical terms by providing an introduction to the AWS Well-Architected tool with an example. By Stefan Christoph, Frank Blessing, and Tufan Özdoğan

An architecture review is a good way to determine whether best practices are being followed in cloud design and, if so, whether your design or approach carries risks. As a cloud provider, Amazon Web Services (AWS) has developed data-based practices that are reflected in the structure and questions of the Well-Architected Framework Review. Over the years – the framework has been in place since 2015 – AWS has formalized matching test mechanisms. A questionnaire helps you apply this uniform compendium of best practices and lets you explore how well an architecture aligns with best practices for the cloud. Additionally, the framework provides detailed guidance on how to eliminate any vulnerabilities you discover.

The goal of the Well-Architected Framework is to achieve a good application design in the cloud that is appropriate for the application's purpose. One such principle, for example, is that of developing environments in a data-driven way. The aim is to measure the influence of architecture decisions and, accordingly, base further development of the

architecture on the facts by collecting metrics for the change process. Each new version of the architecture provides new data points on which organizations can build a continuous evolution process. These data points can then be used in a targeted way to implement improvements.

Six Pillars

The AWS Well-Architected Framework is based on six pillars that look at different aspects:

- Operational excellence
- Security
- Reliability
- Performance efficiency
- Cost optimization
- Sustainability

Operational excellence relates to the design and monitoring of the systems provided. The goals are to generate genuine added value for the business and optimize continuously processes and procedures. Important aspects include automating changes, handling disruptions to operations efficiently, and defining standards for managing day-to-day operations. Building on these aspects are effective

organization of teams and promoting innovation.

Security relates to the security of information and systems. Key areas include confidentiality and data integrity; rights management, including defining and managing individual permissions; protecting systems; and establishing controls to detect security incidents.

Reliability focuses on ensuring that the workload performs its intended function correctly and consistently at the right time. A fail-safe workload ideally recovers quickly from outages to meet business requirements. Important aspects include a distributed system design, recovery planning, and change handling.

Performance efficiency revolves around the efficient use of IT and computing resources. Key issues include selecting the right resources according to the workload, monitoring performance, and making informed decisions on maintaining efficiency as the business grows.

Cost optimization aims to avoid unnecessary costs. The key to success is to establish cost visibility, define budgets, and analyze expenses and

optimize accordingly on a regular basis.

Sustainability is concerned with minimizing the effect that using cloud workloads has on the environment. The key issues include a shared responsibility model for sustainability, understanding environmental impacts, and making the best possible use of existing resources to reduce the effect on the environment.

An Ideal Review

One cornerstone of a successful Well-Architected Framework Review is defining from the outset a concrete scope with a common understanding for the workload to be considered. Making this scope explicit ensures that you can involve the required knowledge holders in the review while marking clear boundaries to other systems. This effort significantly reduces the likelihood of delays during and after the review because the critical workload information can be provided in the course of the meeting. Also, the volume of components to be considered will not grow.

Inviting the business and technical experts to the review meeting is also important because it is the only way to consider all aspects of the workload and to define improvements that result from the discussions held. Although covering all addressed areas of the Well-Architected Framework Review with the participants is elementary, it is equally important to limit the number of participants: A group that is too large can cause important discussions to get out of hand and make the conclusions of the review difficult to implement in a reasonable amount of time. Less is often more.

Discussions in the review meeting are intentional and valuable. Often, such a meeting is one of the few occasions that unites all relevant views. However, it is important to avoid slipping into unproductive and detail-obsessed discussions. Active moderation will help everyone work toward a common goal. Experienced individuals, such as cloud architects who are

not directly involved in the workload and can therefore guide others through the review in an unbiased manner, are often the best choice for moderators.

Make a note of any arguments and additional issues that occur and take them up at a later time. If it is not possible to agree on a common answer to a question, you need to record that fact, too. Often discussions of this type during the review reveal information gaps that can be closed afterward.

Concerns from the Team

Before the review takes place, you should define the purpose for which the review is being carried out. The motivations can be manifold. For example, the team responsible for the workload might want to analyze for itself what vulnerabilities the current workload architecture has. Weak points can often be analyzed quickly in the process. Even in this case, though, you need to make sure nobody involved is afraid to admit their mistakes or is tempted to defend their work or tries to present it in a better light. The easiest way to achieve this is to clarify in advance that the results of the review will only remain within the circle of those directly involved.

Another motivation could be to document the vulnerabilities found and the overhead required to eliminate them, so that additional resources, such as staff or time, can be requested. If communicated clearly, the results are likely to be good. However, it can be very difficult when a review is requested externally (e.g., by management) or is motivated by the desire to provide data points for an external audit. You need to put a great deal of effort into creating an environment of trust, wherein problems can be addressed openly, despite it being a potentially uncomfortable situation. The moderator of such a review needs to question the responses even more critically to achieve the goal of obtaining realistic results from the review.

One important aspect is to orient the participants' mentality. An identified shortcoming – even if it is a high-risk item – does not mean a failure but, instead, an opportunity to make an improvement. It is good to clarify this point in the review itself and not after the risk has occurred, possibly provoking a loss of production.

Adapting the Framework

The goal of the Well-Architected Framework is to provide consistent principles and best practices that can be applied to many different workloads. The general approach addresses the broadest possible range of issues. Not all of them have to apply to the workload under investigation and can therefore be ignored with a comment to that effect, allowing the questionnaire to be customized to the scope applicable to the workload. The results of a review can be varied, and they can be used in different ways. Only in the very rarest of cases is a result likely the perfect implementation of everything, with no further recommendations. Even then, a review is extremely valuable because you now have the assurance that the use case is fundamentally set up well.

The majority of reviews result in identifying a number of risks and proposals for specific solutions and recommendations. This ideal starting point includes tasks in planning for the next releases and their prioritization according to the identified risks.

Translated into the world of agile methods, the result of a Well-Architected Framework Review is a number of user stories that are already well qualified to be transferred to the backlog for prioritization. Figuratively speaking, the team leaves a table full of Post-it notes after the review.

Looking ahead, it makes sense to run the review at regular intervals (e.g., as the workload continues to evolve) or when major changes are made, so you can keep the picture up to date.

The Well-Architected Tool

You can implement the framework directly in the AWS Management Console with the AWS Well-Architected tool by logging in to the management console and opening the tool after signing in to the AWS website [1]. When you use the tool for the first time, you will be taken to a page with an introduction to its functions. The first thing you need to do is to define a workload in one of two ways: by selecting the *Define workload* button in the Define a Workload section, or by selecting *Workloads | Define workload* from the left navigation pane. For details on how AWS uses the data for your workload, select *Why does AWS need this data, and how will it be used?* In the Name field, type a name for your workload. The name must be between three and 100 characters in length. At least three of the characters must not be spaces. Workload names

must be unique. However, spaces and case are ignored when the system checks for uniqueness.

Next, enter a description of the workload in the Description field of between three and 250 characters in length. The Review Owner field asks for the name, email address, or identification key for the primary group or owner of the workload review process. Then, in the Environment field, select the environment for your workload: A *Production* workload runs in a production environment, and a *Pre-production* workload runs in a pre-production environment. In the Regions section, select the regions for your workload. Here you can choose between AWS regions and non-AWS regions; up to five regions (in a comma-separated list) are allowed, and you can use both options if this is meaningful for your workload. The next steps are optional. In the Account IDs box, you can specify the

account IDs related to your workload, and in the Architectural Design field, you can type the URL for your architectural design. The Industry Type field lets you store the type of industry for your workload, and the Industry field is a place for the branch that best matches your workload. In the Tags section, you can further specify tags to be associated with your workload. After that, click *Next*: If a mandatory field is empty or a specified value is invalid, you need to fix the problem before continuing.

Now select the perspectives in the Apply Lenses section that apply to this workload. Up to 20 perspectives can be added to a workload, but this article focuses on the AWS Well Architected Framework perspective. After answering basic questions about your workload, click *Define workload*, and you will be taken to a page with your current workload details. Select *Start review* to begin; otherwise, you

The screenshot displays the AWS Well-Architected Framework 'Questions' page. It is structured into three primary sections:

- Section 1 (Left Sidebar):** A list of security questions (SEC 1-10). SEC 1-3 are marked 'Done'. SEC 4 is the current question: 'How do you detect and investigate security events?'. Below the questions are tabs for 'Reliability' (0/13) and 'Performance Efficiency' (0/9).
- Section 2 (Main Content Area):** Focuses on SEC 4. It includes an 'Ask an expert' button, a description of the question, and several radio button options:
 - ☐ Question does not apply to this workload
 - ☐ Configure service and application logging
 - ☐ Analyze logs, findings, and metrics centrally
 - ☐ Automate response to events
 - ☐ Implement actionable security events
 - ☐ None of these
 There is also a checkbox to 'Mark best practice(s) that don't apply to this workload'. A 'Notes - optional' text area is at the bottom with a 2084 character limit.
- Section 3 (Right Sidebar):** Titled 'Helpful resources', it contains links to various AWS services and documentation related to security and monitoring, such as Amazon GuardDuty, AWS Security Hub, Amazon CloudWatch, and AWS Config.

Navigation buttons at the bottom include 'Save and exit', 'Previous', and 'Next'.

Figure 1: The questions page of the AWS Well-Architected tool is divided into three sections.

can click the *Workloads* option in the left navigation pane and the workload name to open its detail page.

Answering Review Questions

The questions page of the review is divided into three sections (**Figure 1**). The left section shows questions for each of the pillars mentioned at the beginning. The questions that you have answered are tagged *Done*. The number of questions answered for each pillar is written next to its name. You can navigate to questions for other pillars by selecting the name and then selecting the question you want to answer. In the middle pane, you can see the current question where you choose the best practices you follow. Click the *Info* link for additional information on the question or a best practice, or click the *Ask an expert* button to access the dedicated AWS Well Architected section of the AWS Community, re:Post. Additional information and helpful resources appear in the right pane.

When answering each question, use the following procedure:

- Read the question and decide if it applies to your workload (click *Info* for further guidance). If a question does not apply to your workload, select *Question does not*

apply to this workload; otherwise, click on the best practices in the list that you are currently following. If you are not currently using any of these best practices, go for *None of these*.

- If one or more best practices do not apply to your workload, uncheck the methods that do not apply. For each unchecked best practice, you can optionally specify a reason and provide additional details. In general, the Notes field can optionally be used to add information related to the question. For example, you can describe why the question is not applicable or provide additional details on the selected best practices.
- Select *Next* after each question to continue. You can click *Save and exit* at any time to save your changes and stop documenting your workload. To return to the questions, go to the workload detail page and click *Continue reviewing*.

After documenting the status of your workload for the first time, you will want to save the milestone and generate a workload report. A milestone represents the current status of the workload and lets you to measure future progress as you make changes according to your improvement plan. To do this, click the *Save milestone*

button in the workload overview on the workload detail page and type a name (e.g., *Version 1.0 – Initial Review < workload name >*). Do not forget to click *Save*.


If you want to generate a workload report, select the desired view and click *Generate report*. A PDF file is created containing the workload status, the number of risks detected, and a list of recommended improvements.

Conclusions

A review can be challenging because of the volume of questions and organizational barriers in the course of an interdisciplinary exchange. AWS teams or partners can help you with the implementation and act as an independent party to moderate the review and answer questions during the process. Improvements can then be planned in a targeted manner. For better traceability, teams can transfer the improvement potentials as tasks to a ticket management tool, and you'll find it useful to weigh the workload continuously against best practices from the Well-Architected Framework Review. ■

Info

- [1] AWS Well-Architected Tool:
[<https://aws.amazon.com>]
-



Azure Sphere for Internet of Things

Well Rounded

Microsoft Azure Sphere links three vital elements of the Internet of Things - microcontrollers, software, and cloud service - with a focus on security. By Christian Knermann

The Microsoft Azure Sphere platform for integrating microcontrollers with the Internet of Things (IoT) includes both the reference architecture for microcontrollers, matching operating systems, and a cloud service that takes care of updates. In this article I help you get started with Azure Sphere.

Digitization and Optimization

The significant trend of digitization is increasingly determining business life. Economic success often depends on a company's ability to map and optimize processes in production and logistics. IoT and its professional offshoot, industrial IoT (IIoT), belong in this context as subsets under the digitization umbrella. IoT provides a link between information technology and the physical world, creating the basis on which software and artificial intelligence (AI) can be applied. For AI to acquire data in the physical world, measure and draw conclusions from the data, and respond on a physical level, it needs the digital twins of physical assets and software

implementation. Conversely, the machines in the physical world need sensors that collect data and actuators that allow software to intervene.

IoT Elements

Devices that operate according to rules different from the classic client-server model are pushing their way into the network, posing new challenges for IT administrators. Although admins have been confronted with a large number of clients and servers in the past, full-fledged digitization of machines and systems quickly adds a large number of new endpoints.

Typically, these endpoints are microcontrollers (MCUs). The smallest representatives of this genre are devices from the Espressif ESP8266 and ESP32 product families, which have been well used and documented [1]. Systems from the Arduino family are also popular. This MCU platform can be programmed in C++ in its integrated development environment (the Arduino IDE), and both the hardware and software are available under an open source license.

The controller is usually installed on a development board, connected directly by USB to a system with an installed IDE. A board like this combines processor, memory, and timer components; a digital-to-analog converter; and connectivity by a wireless local area network (WLAN) or Bluetooth. To further expand functionality, expansion boards known as hardware attached on top (HATs) or shields add sensors, actuators, or small displays to the board. Grove expansion boards are also available as an alternative. A Grove shield routes the MCU's connections out to simple slots so that all other components can be connected with standardized plugin cables and without soldering.

MCUs try to connect to the network over WLAN or a low-power, wide-area network (LoRaWAN) over longer distances. At the logical data transfer level, don't forget legacy HTTP, on which representational state transfer (REST) enables communication to and from IoT devices by common HTTP methods such as GET and POST. Protocols such as the Advanced Message Queuing Protocol (AMQP) and Message Queuing Telemetry Transport

Photo by Zosia Korcz on Unsplash

(MQTT) protocol are still very common IoT methods. MQTT has minimal overhead and is optimized to allow low-powered devices to transmit data reliably, even over unreliable networks. MQTT follows the publish-subscribe pattern. Clients can send messages with a certain topic, and other clients subscribe to them in a style known from social networks. The communication hub is an MQTT broker that receives the data and takes care of distributing messages in push mode.

Azure Sphere for Enterprise IoT

Because they are easily accessible, Arduinos and comparable MCUs are popular in the consumer space, but developing on this platform poses challenges once a project needs to scale to production levels. What do you need to consider if you are no longer working with a few prototypes, but a population of hundreds or even thousands of endpoints on your network that can both measure and influence business-critical processes? In this case, comprehensive distribution of firmware and application updates to the endpoints must be ensured, and securing communications, authentication, and authorization is equally important. This time-consuming activity usually involves dealing with security strategies such as certificate-based encryption, and unfortunately, it is all too often neglected or not taken seriously enough on the fast path from development to production. Microsoft enters the scene for these situations with Azure Sphere [2]. With Azure Sphere, Microsoft creates its own platform for IoT that includes far more than just MCU hardware. Azure Sphere links three vital elements: hardware, software, and cloud service. Safety is not an add-on that needs to be retrofitted at some later stage, but an integral part of the platform right from the outset.

Certified Hardware

The first important building block is an MCU that is certified for Azure

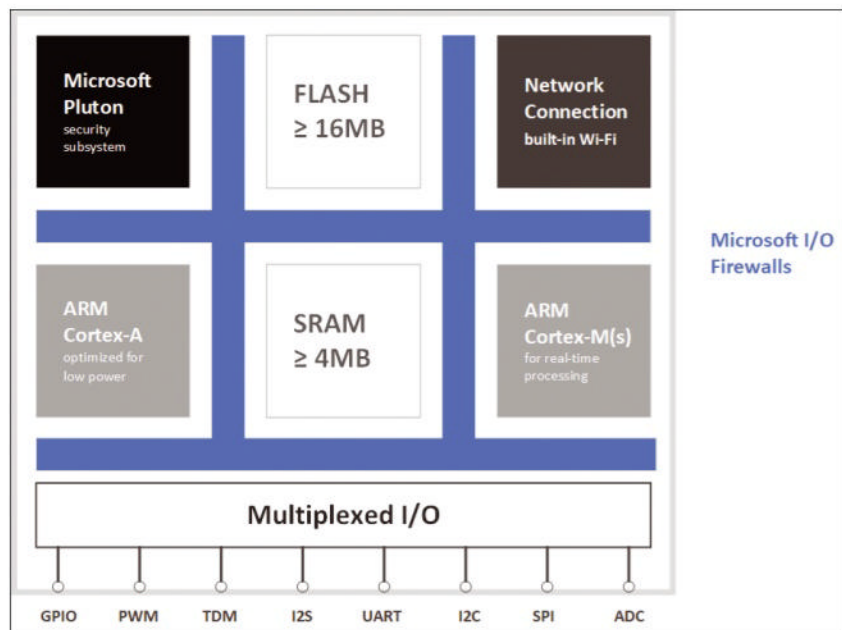


Figure 1: The MediaTek MT3620 microcontroller has five CPU cores and SRAM. © Microsoft

Sphere. Microsoft does not manufacture this kind of hardware but works with third-party suppliers that manufacture devices to spec. Several compatible devices from various manufacturers can be found on the market [3]. The core of all devices is the MT3620 processor by MediaTek with an ARM32 processor architecture and a total of five cores (Figure 1). One of these cores is Microsoft's Pluton security chip, which is basically comparable to a Trusted Platform Module (TPM), although it offers

significantly more functionality. Pluton acts as a hardware root of trust and random number generator, securing the MCU with Secure Boot and implementing encryption functions. The chip makes sure a device does not execute unsigned code, and the integrity of each device is maintained with remote attestation in collaboration with the Azure Sphere cloud service. Each MCU is uniquely identifiable worldwide for this purpose. The MCU also has an ARM Cortex-A core, which is designed for

Table 1: MCU Interfaces

Interface	Function
GPIO (general-purpose input/output)	An interface for bidirectional transmission of digital signals, whose function is completely definable in software. One pin each maps the values 0 and 1 in the simplest case – for example, to switch on an LED (output) or to query the position of a switch (input).
PWM (pulse width modulation)	Generates variable analog signals (e.g., suitable for controlling motors or the brightness of LEDs).
TDM (time-division multiplexing)	A digital interface for transmitting multiple data streams in a single signal.
I2S (inter-integrated sound)	A digital serial bus for transmitting audio signals.
UART (universal asynchronous receiver/transmitter)	A bidirectional serial interface for communicating with connected devices (e.g., for a classic terminal connection with a PC connected by USB for debugging purposes). Some sensors communicate over UART.
I2C (inter-integrated circuit)	A bidirectional serial communication similar to UART typically used for modules and sensors.
SPI (serial peripheral interface)	Bidirectional serial communication; faster than UART and I2C.
ADC (analog-to-digital converter)	Converts analog signals into digital signals (e.g., for sensors that measure temperature, humidity, or electrical voltage).

particularly low power consumption, and two ARM Cortex-M cores. The latter are optimized for real-time control functions. The MCU has a variety of interfaces and devices for interaction with sensors, actuators, and other peripherals and can exclusively assign all ports for peripherals (Table 1) to one of these cores and ensure that code executed by this core cannot access the other cores. Finally, a separate core implements the WLAN subsystem. The MCU is dual-band and compliant with the 802.11a/b/g/n standards. The MT3620 isolates the security functions and the WLAN from end-user code.

Linux and Local Apps

Certified MCUs come with the Azure Sphere operating system out of the box. It is a Microsoft-customized and open source Linux system with a hardened kernel specifically optimized for IoT applications. The operating system is reduced to the bare essentials and has neither a shell nor a package manager. You can either feed in application code in the form of a compiled image to a local device by USB or deliver your images in the cloud with the Azure Sphere Security Service (firmware over the air, FOTA), which is the approach of choice for larger numbers of devices, especially because endpoints in production are often geographically distributed and permanently installed in systems, so no longer physically accessible. Transferring images directly by USB is also known as side-loading, but this method only works if the respective device is registered with the cloud service. Microsoft distinguishes between high-level apps that run on the A-core of the MCU and low-level apps that use the M-cores [4]. A high-level app runs in the user context and can only use defined libraries and API functions. It mediates certificate-based, authenticated connections between devices and the cloud, interacts with interfaces such as GPIO or UART, and communicates with low-level apps that access the hardware directly or

by an additional real-time operating system (RTOS). Each low-level app runs in complete isolation and cannot interact with the outside world directly, but only with a high-level app.

Cloud Service Updates

The Azure Sphere Security Service takes care of remote attestation, ensuring that the MCU is genuine and tamper-free with a bona fide and up-to-date operating system. The cloud service uses authentication with client certificates to secure device-to-device and device-to-cloud communication. Devices only need to contact the cloud service for updates of the operating system and application images. Apart from that, the devices also work in offline mode. The operating system and cloud service are included in the purchase price of the MCU. With each controller, you get 10 years of support and updates for the operating system and the right to use the cloud service for the same amount of time. Beyond that, you won't incur any additional costs – unless you use Microsoft's in-house Azure IoT services in conjunction with Azure Sphere devices, for which Microsoft will bill you separately. However, Azure IoT Central and the Azure IoT Hub [5] are optional. Although the Azure Sphere Security Service

runs within the Azure Cloud, you can integrate your Azure Sphere devices with any service in a public or private cloud for control tasks and further processing of data, such as sending data by MQTT to a broker on your local network.

Development Boards

Most of the sample code for newcomers relies on the Azure Sphere MT3620 Development Kit by Seeed Studio [6]. This development board comes with two integrated WLAN antennas and two connectors for external antennas. It also has two function keys, a reset button, several status LEDs, and a micro-USB port that provides power and an interface for programming and debugging (Figure 2). The connections for peripheral devices are accessible by two rows of double pins, or headers. Together with the Grove Starter Kit from the same manufacturer, the development board forms a solid basis for first steps [7]. However, because of its layout, it is unfortunately not compatible with existing shields for Arduino systems. Microsoft guides you through the commissioning process with easy-to-follow instructions [8]. You can choose whether to develop images for the device with the full-blown Visual Studio IDE or the versatile Visual

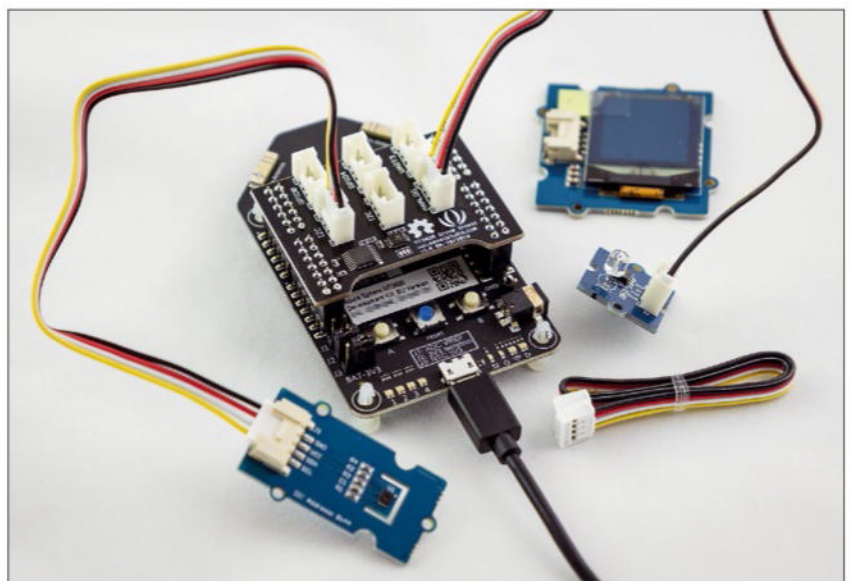


Figure 2: The Grove Starter Kit extends the Azure Sphere MT3620 Development Kit without the need to solder.

Studio Code editor. If you opt for the IDE, the Community Edition is suitable, although Microsoft only offers it free of charge to students, genuine open source projects, and individual developers. In contrast, the Visual Studio Code editor, which I also used in the tests, is available free of charge for commercial projects and enterprises of any size.

Regardless of which development environment you prefer, the first step is to install the Azure Sphere Software Development Kit (SDK) with the

```
azsphere
```

command, which runs both on the classic command line and in PowerShell sessions [9]. Also, according to Microsoft's documentation, you need to install Visual Studio Code, followed by the additional CMake and Ninja packages before launching Visual Studio Code with the included extensions for Azure Sphere.

Connect your development board to the computer by USB. You can then communicate with the device with the `azsphere` command, which, thanks to the built-in license, entitles you to create a new customer account, or tenant, with Azure Sphere Security Service and to register the device. In the case of the development board in this example, however, I first had to reinstall (recover) the operating system because the version that came with the board was simply obsolete:

```
azsphere register-user 2
--new-user <your mail address>
azsphere device recover
azsphere tenant create --name <your name>
azsphere tenant select --tenant <uid>
azsphere device claim
```

You are free to choose the name of your tenant, but be aware that the subsequent claim irreversibly connects the device to your tenant. Security is the top priority here. The MCU does not accept images outside of your tenant during a cloud deployment, which rules out hijacking by another tenant.

First Signals from the Cloud

Before you go any further, you need to enable the device to run code locally. Microsoft's learning path then takes you to the simplest imaginable example. Whereas budding programmers in almost any language first learn to output "Hello World," the equivalent in the field of microcontrollers is to make an LED flash on the device. Microsoft also provides some sample code; you need to compile the code in Visual Studio Code, copy it to the MCU over USB, and execute the code [10].

A cloud deployment, in comparison, is far more exciting because it lets the Azure Sphere cloud service play to its strengths. You create a "product" in the cloud as a management unit for a specific application [11]. You then assign one or more MCUs to the product, and – assuming the WiFi configuration is correct – it will retrieve images of your applications from the cloud with FOTA from then on. For this to happen, the Azure Sphere service automatically creates device groups for each product, including development, field test, and production, giving you the best possible support for the development process.

Conclusions

Although conventional approaches to development for IoT often, unfortunately, mean security falling by the wayside or being painstakingly retrofitted to the finished product, Microsoft has thought about protecting the entire platform right from the outset with Azure Sphere. Microsoft's approach of anchoring security in the hardware of the MCUs and linking it to a cloud service addresses professional use cases and ensures that you can focus entirely on your application logic. Azure Sphere takes care of the framework and provides the required infrastructure for a staged process from development to testing to production.

You can already find Azure Sphere-compatible hardware on the market

and numerous code samples on GitHub. The only downside is the lack of compatibility with the Arduino ecosystem. This platform offers a large range of hardware and application examples, making it easier to get started, but it is less widespread in the professional environment outside of the maker scene. ■

Info

- [1] Espressif MCUs: <https://www.linuxpromagazine.com/content/search?SearchText=ESP8266&x=0&y=0>
- [2] Azure Sphere: <https://azure.microsoft.com/en-us/products/azure-sphere/>
- [3] Hardware certified for Azure Sphere: <https://devicecatalog.azure.com/devices?searchTerm=azure%20sphere>
- [4] Applications in Azure Sphere: <https://learn.microsoft.com/en-us/azure-sphere/app-development/applications-overview>
- [5] Azure IoT Hub: <https://azure.microsoft.com/en-us/products/iot-hub/>
- [6] Azure Sphere MT3620 Development Kit: https://wiki.seeedstudio.com/Azure_Sphere_MT3620_Development_Kit/
- [7] Grove Starter Kit for Azure Sphere MT3620 Development Kit: https://wiki.seeedstudio.com/Grove_Starter_Kit_for_Azure_Sphere_MT3620_Development_Kit/
- [8] Getting Started with Azure Sphere: <https://azure.microsoft.com/en-us/products/azure-sphere/get-started/>
- [9] Quickstarts for setting up your Azure Sphere device: <https://learn.microsoft.com/en-us/azure-sphere/install/overview>
- [10] Creating a generic application: <https://learn.microsoft.com/en-us/azure-sphere/install/qs-blink-application?tabs=windows%2Ccliv2beta&pivots=vs-code>
- [11] Creating a cloud deployment: <https://learn.microsoft.com/en-us/azure-sphere/install/qs-first-deployment?tabs=cliv2beta>

Author

Christian Knerrmann is Head of IT-Management at Fraunhofer UMSICHT, a German research institute. He's written freelance about computing technology since 2006.





Test your system to help fight phishing attacks

Phish Food

The Gophish phishing framework lets you set up your own phishing campaigns to identify vulnerabilities and make users aware of these dangers. By Holger Reibold

Safeguards for the IT infrastructure often neglect email as an attack vector. Although most companies run a spam filter, they pay far too little attention to phishing, and many companies have already fallen victim to data theft, espionage, and sabotage. The industry association Bitkom estimates the annual damage to German institutions by these attacks, which are carried out in an increasingly professional manner, at more than EUR 200 billion (~\$163 billion) [1].

Although most companies focus their security campaigns on hardening their own infrastructures, they overlook the fact that the real threats lurk elsewhere: 85 percent of cybersecurity breaches are due to human error, and 94 percent of all malware finds its way to its recipient by email. More than 80 percent of security-related events are phishing attacks. Attackers have long since stopped focusing on seemingly attractive corporations and large companies and are increasingly targeting small and medium-sized enterprises (SMEs), which are targeted by attackers precisely because they invest significantly less in their security architecture, whether by choice or because of budget restrictions.

The consequences of these findings is that companies need to invest more in their email security; in particular,

protection against phishing attacks need significant improvement. This is where phishing penetration testing comes in: Gophish [2] provides an open source framework for precisely this task.

Gophish at a Glance

In view of the huge relevance of the phishing problem and the associated threat situation, surprisingly, most companies rely on extensions of established filter programs (e.g., SpamAssassin) that typically use plugins to combat phishing. However, it is not enough to filter out critical messages; instead, IT managers need to check their own infrastructures for vulnerabilities. Dynamic environments and temporarily logged-in clients such as field workers' notebooks, tablets, and smartphones pose a particular challenge.

The Gophish framework lets you simulate phishing attacks, enabling phishing training for any type of organization. Gophish is written in the Go programming language, and the central benefit is that the compiled binaries do not have any dependencies.

Getting Started

You can simply download and run the software – no installation required.

In the case of a source code-based installation, you need to configure the interaction with a MySQL server, and you also need SSL certificates and private keys. Finally, you need to make various adjustments (e.g., to the IP address and port configuration) in the `config.json` file in the root directory of the Gophish installation. Compiled packages for Linux, macOS, and Windows are available for download [3]. To get started, simply unpack and start the Gophish server.

Penetration testers usually turn to Kali Linux for their work, although Gophish is not preinstalled in Kali. To install, download and unzip Gophish to a directory of your choice, and then assign the required permissions:

```
chmod +x gophish
```

Configuration is adjusted by editing the `config.json` file. In addition to the IP address, you need to specify the paths to the SSL keys and certificates. To start the application, type:

```
./gophish
```

The implementation of a phishing campaign comprises three steps: (1) generate the templates and determine the targets, (2) send the phishing email on its way – staggering the timing between messages, if

necessary, (3) track the results, which Gophish visualizes in real time on its dashboard.

Preparing a Campaign

On Windows you launch the environment by running `gophish.exe`; on macOS and Linux, use the binaries for your choice of OS. By default, the web interface can be accessed on `https://127.0.0.1:3333/`. The username is `admin`, and the password is output on the console. Before you can access the Administration Center, you need to define a new password, after which, the environment is at your disposal. During the initial installation, Gophish tells you that you don't have a campaign yet.

The first step is to create a sending profile by switching to the Sending Profiles menu and creating an initial configuration in *New Profile*. The description in [Figure 1](#) uses a virtual machine (VM) on 192.168.178.100 and is the sender of the phishing email. Now assign the typical data for sending email to this sender, which will then wait for messages on the specified address. It is important that you use a valid send port. You can also use a custom header.

If you are just starting out, it is a good idea to send a test email to check the functionality by clicking *Send Test Email*. A click on *Save Profile* lets you save the initial profile configuration.

Before launching a phishing campaign, you need to define the targets, for which you can use various tools. If you want to collect email addresses from public information to simulate as realistic a scenario as possible, you can turn to open source intelligence (OSINT), for example. To test the local infrastructure, you need the local email addresses. Regardless of the data source, you need to create an initial group in the Users & Groups menu by clicking on *New Group*. Assign a name and save the addresses of your target group. The easiest way to do this is to use the bulk import function with a CSV file. For the import to work, it needs First Name, Last

Name, Email Address, and Position header values. You can also create some test receivers manually. Click *Save changes* to save the group.

Creating a Template

Once the recipients of your first phishing campaign are created, it's time to create a template, which is the actual phishing email or, more precisely, its content. To do this, go to the Email Templates menu. You can import existing email content with the *Import Email* button. One typical attack vector in phishing email is to get the recipient of the message to reset the password. To this end, the templates provide various functions. Gophish also provides various variables for use in the email templates and

on the landing page ([Table 1](#)). Note that the software is case-sensitive for templates. In the Subject line, use the following configuration to contact all recipients from an email address pool:

Reset password for {{.Email}}

In the text field you can then start entering the message content, which you design in the *HTML* tab. Gophish also has a simple visual editor, which you open by clicking the *Source* button. To illustrate how the template works, enter the following text in HTML view:

```
Hello {{.FirstName}},
Your password for {{.Email}} has expired. 2
Please request a new password here.
Kind regards,
Your Support Team
```

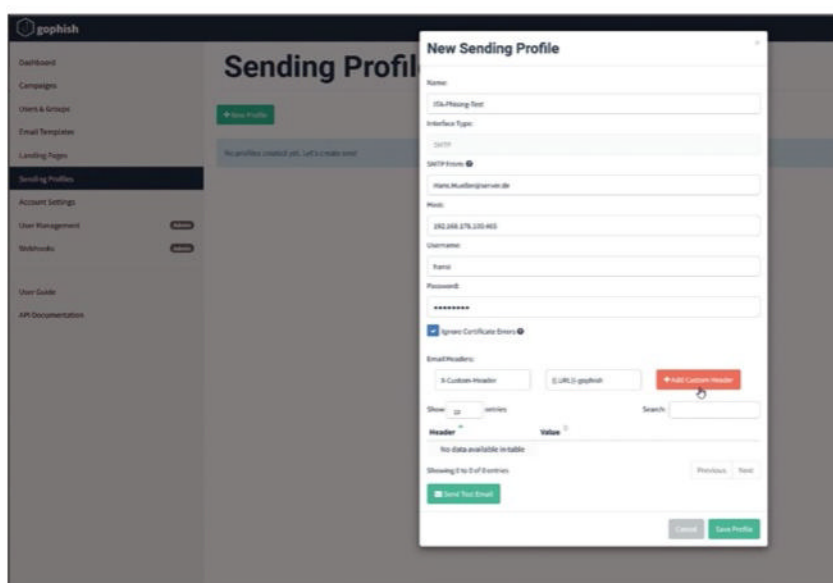


Figure 1: The phishing test starts by creating a sender profile.

Table 1: Supported Variables

Variable	Function
{{.Rid}}	Unique ID of the target
{{.FirstName}}	First name of the target
{{.LastName}}	Last name of the recipient
{{.Position}}	Role
{{.Email}}	Email address
{{.From}}	Fake email address of the sender
{{.TrackingURL}}	URL for the tracking handler
{{.Tracker}}	Alias for <code></code>
{{.URL}}	Phishing URL
{{.BaseURL}}	Base URL without the path and routine identifier (RID) parameters, which is useful for creating links to static files

You now need to serve up a here link to the message recipient by selecting the word *here*, clicking the chain icon, and assigning the URL link type `http://` as the protocol and the target URL for the link on the *Link Info* tab. Instead of a fixed URL, you again need to use a variable; this time it's `{{.URL}}`. This configuration ensures that you can assign individual URLs to different campaigns. Make sure that *Add Tracking Image* is checked to ensure user tracking.

Grabbing Passwords

The goal of a phishing email is to gain personal information. The primary aim is to steal access credentials and use them for further attacks. To do this, attackers lure their victims to websites that are often perfect replicas of the purported target sites. Therefore, you also need a portal for your phishing test. You can set this up from the Landing Pages menu.

This step is quite simple. Clicking *New Page* lets you generate a copy of the page whose URL you store by selecting *Import Site*. In this example, I simulate access to the admin area of a web-based enterprise resource planning (ERP) installation. After the import, the HTML code of the page shows up and a click on the *Source* button provides a preview.

To record the forms the victims submit, check the *Capture Submitted Data* and *Capture Passwords* options. This input is not encrypted and is stored in the Gophish database in plaintext. Once the victim has given you this information, you can redirect them to another page that confirms, for example, that the password update has been changed, so that the victim feels secure. You can specify the redirect URL in the *Redirect to:* input box. A final click on *Save Page* saves the target page.

Running a Campaign

Preparations are complete for a first launch of your first phishing campaign. To do this, switch to the

Campaigns menu and create an initial campaign by selecting *New Campaign*. Most of the settings are self-explanatory: Assign a name for the campaign, select the email template, and determine the landing page. The configuration of the URL input field is where you specify the IP address of the Gophish server. It is important for the server to be available during the campaign so it can track and record the client actions.

The campaign configuration supports time control in the *Launch Date* input box. Next, specify the channel profile (*Sending Profile*) and the target audience (*Groups*) and click *Launch Campaign* to launch your first test (Figure 2). After starting the campaign, you will be automatically redirected to the results page where you can track email sending and opened messages in real time (Figure 3). The visualization shows the number of email messages sent and opened and the number of email messages from which the target person followed the link and submitted the input form.

In the Details section, the dashboard lists information such as the name of the target and the respective status. The status column tells you which employees fell for the phishing email.

From this information, you might be able to identify patterns in successful attacks and derive consequences for adapting the infrastructure. Making employees aware of the problem once again would be prudent.

The underpinnings of the report function come courtesy of the GoReport [4] module and include export options for downstream processing. Just follow the *View Results* link in the dashboard at the end of the respective test configuration. In the results view, GoReport lists the details of a campaign. From the detailed view, you can export the results or the raw data to a CSV file. The Gophish API is available for custom reports (e.g., to bundle results from multiple campaigns). The developers provide a Python API client to implement appropriate functions. To end a campaign, execute the *Complete* command in the results overview. The current Gophish version does not support automatic termination of campaigns.

Receiving Reports by Email

If you have carefully monitored the console output during the Gophish startup process, you are probably aware that the IMAP manager started

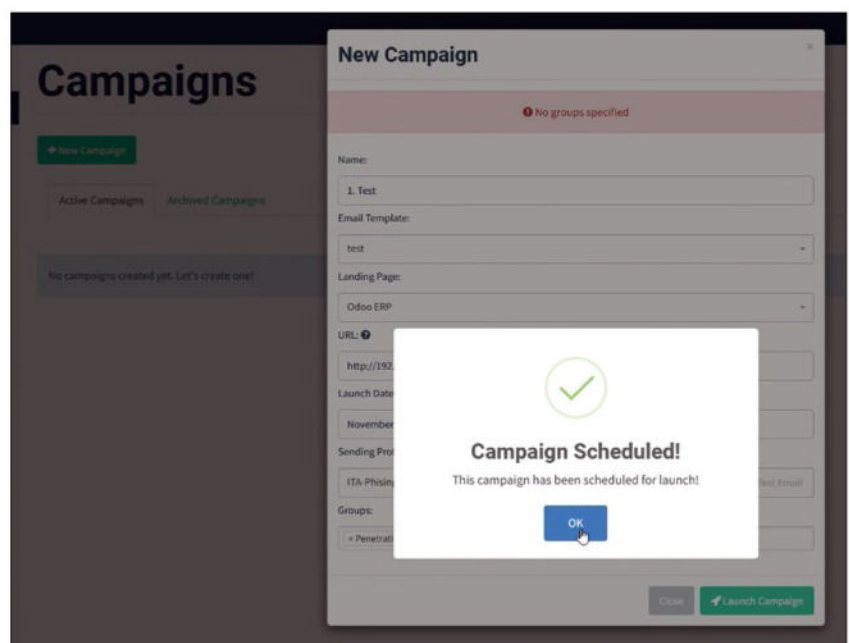


Figure 2: After you launch a campaign, the copy of the phishing landing page will look very similar to the default login page.

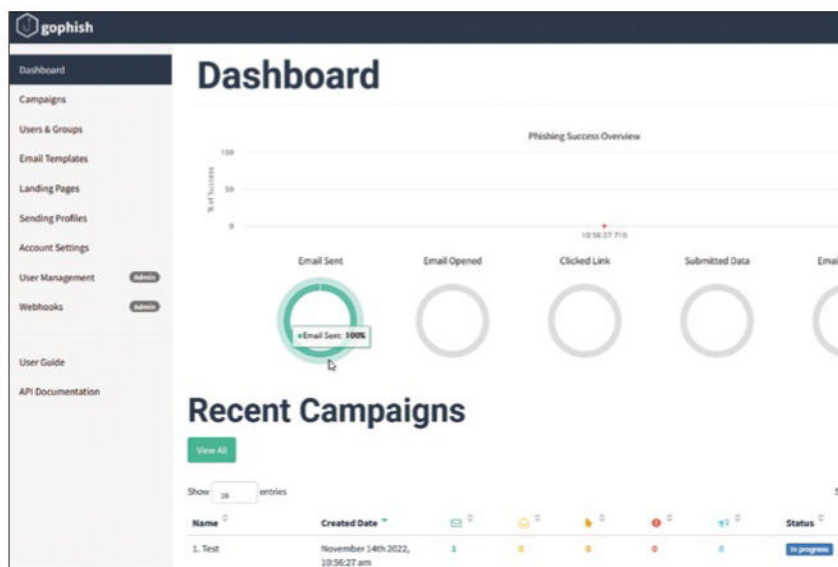


Figure 3: The Gophish dashboard gives you real-time phishing campaign results.

up. Because Gophish does not yet have its own function for sending reports by email, the developers decided to implement a reporting mechanism. Ideally, only a small number of users will have fallen for the fake email, but the administrator can only take action if notified of these incidents. To use the Gophish reporting mechanism, you need to set up an email address that will receive the relevant notifications.

Gophish offers the option of accessing an IMAP mailbox. Once it identifies a campaign email, the software reports this result. You first need to configure the IMAP settings for each Gophish user in the *Account Settings* | *Reporting Settings* option. The *Advanced Settings* button lets you determine the folders and the polling frequency. You can also check the configuration by clicking *Test Settings*.

Email Attachments

One genuine highlight of Gophish is the *Attachment Tracking* function that lets you add attachments with DOCX, DOCM, PPTX, XLSX, XLSM, TXT, HTML, and ICS file types to Gophish templates. When a campaign is launched, the variables defined in these documents are replaced with the matching values. The benefits are obvious, because, for example, in the case of an Office attachment, you can

determine whether a victim opened it: When a prepared document is opened, the Office application tries to load the image, and this access attempt is then registered by the Gophish server.

To begin, create a Word document and insert a module by clicking *Quick Parts* on the *Insert* tab in the *Text* group. Select *Field* and enter `{{.TrackingURL}}` in its properties, and in the field options enable the *Data not saved in document* option. To use the first and last name variables in the Word document, you need to disable the grammar and spelling checker, otherwise Word will register an error. Gophish can also register macro execution with this pattern. In the template settings, attach the modified document.

Managing Gophish

The internal admin functions are limited to user management, web-hook configuration, and logging. In addition to the admin user you created when Gophish went live, you can create users by selecting *User Management* | *New User*. In the associated dialog, assign a role, the username, and a password. You can choose between the admin and standard user roles. The current version does not envisage the addition of more roles.

Basically, Gophish retrieves results through an API. In practice, though, it is often desirable for updates to be reported immediately after an event is registered. Gophish solves this problem by providing webhook support. In a webhook configuration, Gophish sends an HTTP request to a specific endpoint – the request can be signed if required. The request contains the JSON text of the currently registered event, which can then be processed downstream in a third-party application. The webhook configuration is set up in the menu of the same name.

The logging functionality is fairly rudimentary. By default, the logs are output to the standard error output (stderr). If you want the logs to be written to a file, use the command:

```
gophish.log 2>&1
```

You can also use an external security information and event management (SIEM) system in this way.

Conclusions

Gophish offers a new dimension in the fight against phishing email. The phishing penetration testing environment helps you approach this problem from the attackers' perspective and draw conclusions on the optimization of security structures from the insights gained. Despite certain limitations, Gophish offers significant added value. ■

Info

- [1] German businesses under attack: [\[https://www.bitkom.org/EN/List-and-detailpages/Press/German-business-losses-more-than-220-billion-euros-per-year\]](https://www.bitkom.org/EN/List-and-detailpages/Press/German-business-losses-more-than-220-billion-euros-per-year)
- [2] Gophish: [\[https://getgophish.com\]](https://getgophish.com)
- [3] Gophish on GitHub: [\[https://github.com/gophish/gophish/releases\]](https://github.com/gophish/gophish/releases)
- [4] GoReport: [\[https://github.com/chrismaddalena/Goreport/x\]](https://github.com/chrismaddalena/Goreport/x)

The Author

Holger Reibold, computer scientist, has worked as an IT journalist since 1995. His main interests are open source tools and security topics.

Hone your skills with special editions!

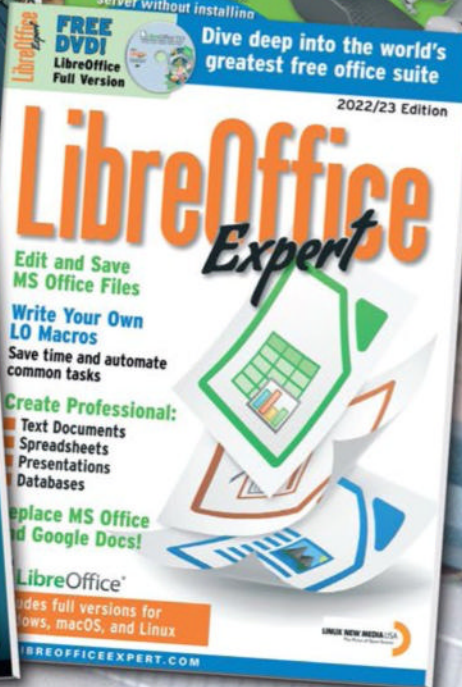
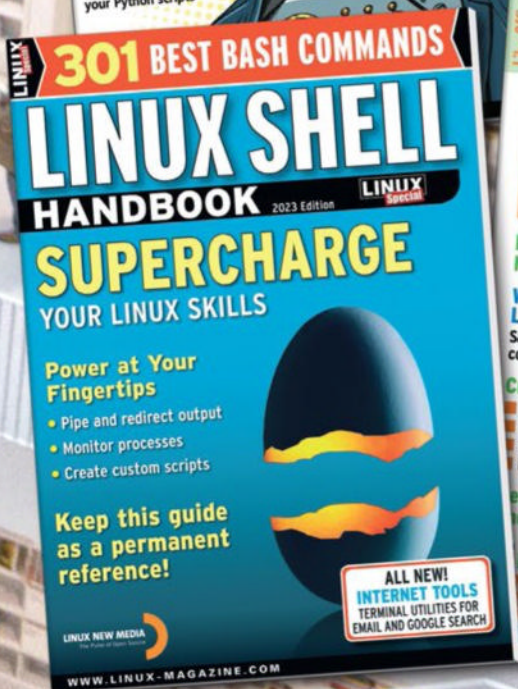
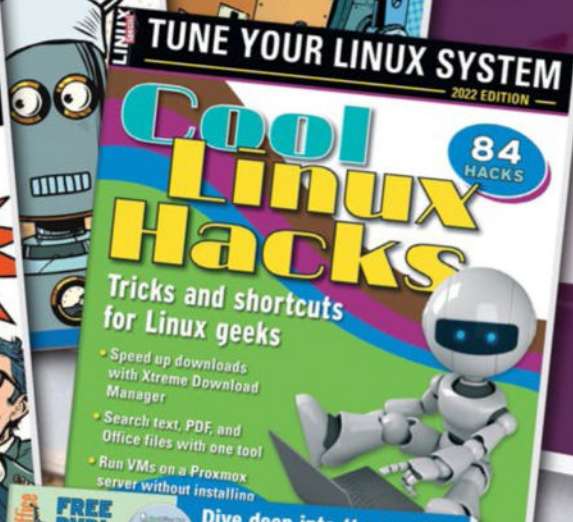
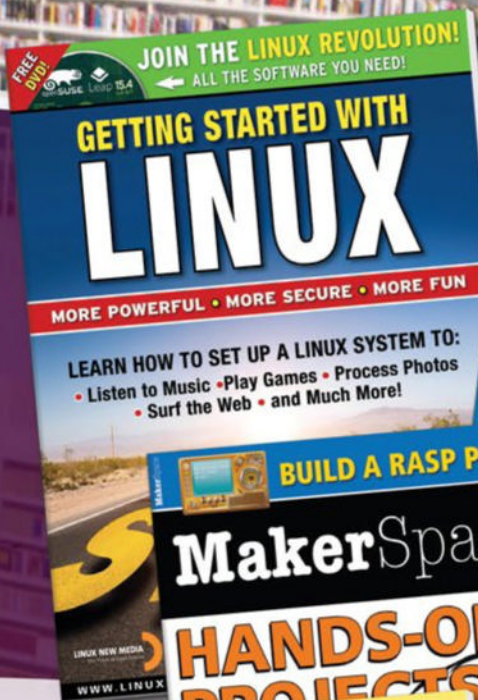
Get to know Shell, LibreOffice, Linux, and more from our Special Edition library.


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

Available in print or digital format

Check out the full library!
sparkhaus-shop.com/specials







Synchronize passwords in KeePass

Digital Safe

Username and passwords play an important role in security. In this article, we show you how to set up the KeePass password manager and keep it synchronized across multiple devices. By Thomas Joos

Password management tools such as KeePass [1] are hugely helpful when dealing with access credentials. The secure and encrypted database of the free, open source software lets you store your login credentials, including notes, links, and other information you need for access. To use this information, you only need the password for the password safe itself, making it far easier to manage passwords, especially when complex combinations are used. The benefits KeePass offers are great even if you only run it on a single computer.

However, the tool is not only designed for single-computer use; you can use the database to store your credentials on multiple devices. The associated file is only a few kilobytes in size, even if it contains many entries, and the content is securely encrypted by 256-bit AES. KeePass clients for many operating systems – besides Linux, macOS, and Windows – include Android, iOS, and iPadOS.

In addition to keeping your passwords safe, you'll benefit from more

convenient and easier logins to various services, because you'll be able to copy the credentials and passwords to the clipboard.

Database Entries

Of course, synchronizing data from KeePass with other services always starts with installing KeePass on a computer, creating a new database, and then adding some initial entries. You can create several databases with the tool and use the one you currently need with the program. If you want, you can change the language in KeePass after the install by selecting *View | Change Language*; then, download the appropriate language files in the window of the download page and (on Windows) copy them to the directory `C:\Program Files\KeePass Password Safe 2\Languages`. To change the language, double-click. If you install KeePass on multiple computers, you can simply copy the language file. You only need to download it once. From the context menu of your

database in KeePass, use *Add Group* to create a new password group. Groups are basically equivalent to folders that help you organize your credentials. In the window, you specify the name of the group and define the icon to be displayed. The name and icon can be adjusted at any time by opening the context menu and selecting *Edit Group*. Once you have opened a group, you can create a new entry by selecting *Add Entry* in the context menu for the right window (Figure 1). Enter the name (*Title*) as you want it to be displayed in KeePass, along with the login name (*User name*) and password (*Password* and *Repeat*) for that entry. You can press the button with the three dots next to the password to show and hide the password. You also have a place to enter a description or supplementary information, such as an associated account number, for the entry (*Notes*). In this field you can freely choose the data to enter. PINs, TANs, and other information are in safe hands in KeePass, as are tax numbers, account

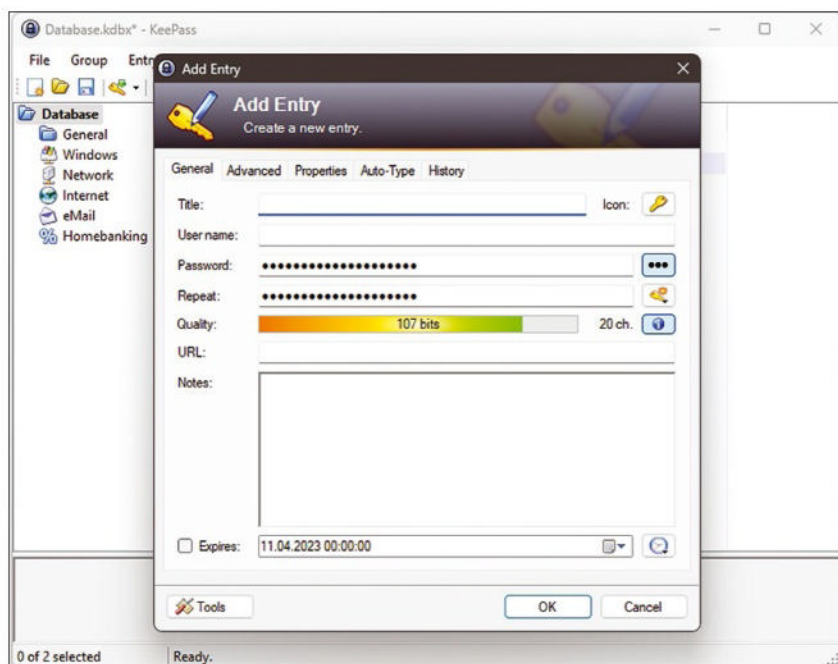


Figure 1: KeePass lets you store your passwords and critical data in a central interface. Folder structures and various databases help organize your data.

numbers, and other data you need all the time but want to keep safe. Admins can even store SSH keys and use them directly from KeePass with the KeeAgent plugin. The Remote Desktop Manager Plugin extension lets you use credentials for Microsoft Remote Desktop Manager. I will go into more detail about plugins in a separate section.

Once you have created your first password database, launch KeePass and open the database, enter the database password, and navigate to the entries you need; you can store these in a tree structure. KeePass remembers the database you last opened and opens it automatically the next time you start the program. To control this behavior, go to *Tools | Options* and select the *Remember and automatically open last used database on startup* entry in the *Advanced* tab.

This option is useful if you work with different account names and different databases on a single computer. If you click on an entry in the database, KeePass displays the associated data. If you have entered a URL for the entry, you can open the website directly in the default system browser. KeePass also gives you a search field where you can search for entries. This option is especially interesting if you work with numerous data records and have forgotten where a particular record is stored.

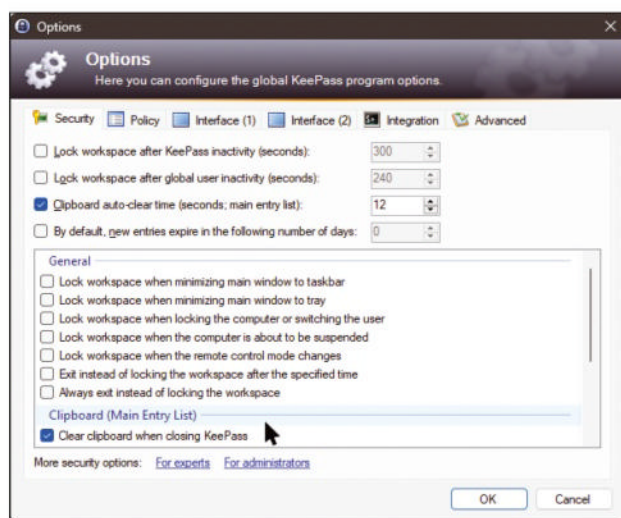


Figure 2: The KeePass options let you customize the solution to suit your needs.

Double-click on the *Password* column to copy passwords to the clipboard. Windows only stores the password in the clipboard for a few seconds and then deletes it again for security reasons. You can set the number of seconds the password is available in the clipboard in *Tools | Options | Security* by selecting *Clipboard auto-clear time* (Figure 2). The context menu and various paste options can be used to paste passwords into different programs.

KeePass on Mobile Devices

Compatible apps (e.g., IOSKeePass or KeePass Touch) let you use your KeePass data on the iPhone or iPad. For Android, you can use the KeePass2Android, KeePassDroid, KeePassMob, or KeepShare apps (Figure 3). It is worth testing the various apps because they offer

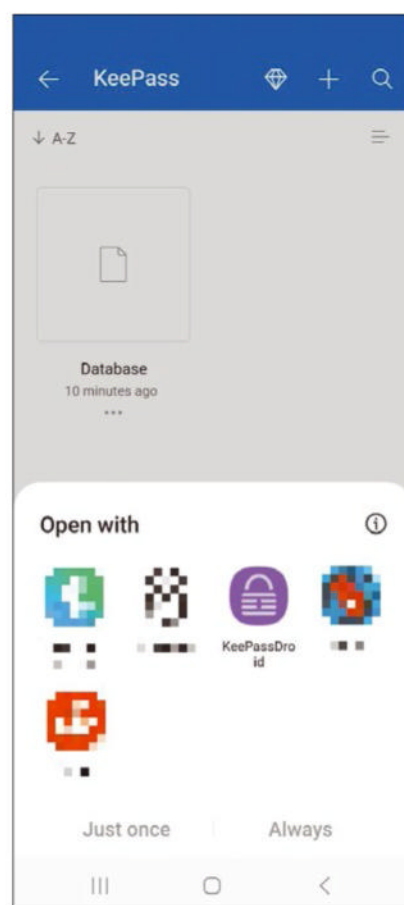


Figure 3: With numerous apps on the iPhone and iPad and on Android devices, you can access KeePass password databases on the go if you store them in the cloud.

different features, such as direct access to cloud storage. Just make sure you choose an app that can handle your version of the database. For mobile access, the database needs to be available on your smartphone. Access to a password database is possible with cloud storage synchronization. You can use pretty much any kind of cloud storage and download the database to or synchronize it on the mobile device. Note that changes might only change a copy of the database, not the original. You always need to check whether you are opening a database in the cloud or on the local device; this is especially important when it comes to adding or editing entries.

The mobile apps basically offer the same capabilities as the KeePass desktop app. For example, on iPhone, iPad, or Android, you can go to the Dropbox app and open the KeePass file. This procedure works with virtually any kind of cloud storage. You can download files in different ways, open them, and then select the KeePass app you want to use to open the database. To keep the data on the mobile device up to date, make sure that the password database on the PC is constantly synchronized to cloud

storage to ensure that the database in the cloud is always up to date.

With some work, you can also synchronize the data with a private cloud. However, store the database file on the local network (e.g., Nextcloud, ownCloud, or a similar solution), not in a public cloud. You can also save the data on a Fritz!Box or network-attached storage (NAS) or on a similar device that allows mobile access. The only important feature is that you are able to access the storage on the other devices.

The various apps for iOS and Android also partially support direct access to cloud storage without having the storage client on the terminal device. This scheme can make it easier to work with the database because you don't have to download it and open it in the app first; rather, you access the database in the cloud directly from the app. The KeePassDroid Android variant supports OneDrive and Google Drive, for example (Figure 4). These two services offer a particularly easy option for transferring data to a smartphone or tablet. You need to experiment to find the combination of cloud storage and smartphone app that is the most convenient to help you share KeePass data.

Helpful Plugins

In parallel, you can download special extensions for the application in KeePass on Windows by selecting *Tools | Plugins | Get More Plugins*. Doing so means that you can upload data directly to your cloud storage. Examples of this can be found in the *I/O & Synchronization* section on the website [2]. Some experimentation is required at this point. Try out the plugins to see how they work. The extensions are available as PLGX files, which you integrate by copying the file to the plugins directory in KeePass. The program automatically loads the plugins on restart; they usually turn up in the Tools menu, where you can configure how you want KeePass to synchronize the database. Often, the plugins connect directly to your cloud storage; you only need to activate the connection once because KeePass accesses an API belonging to the respective cloud service with its plugin.

However, the easiest approach is to upload the KeePass database to the cloud with the synchronization client for your choice of cloud storage and send it back to your smartphone

or tablet from the appropriate cloud storage app. In this way, you can access the data and keep track of synchronization easily. Plugins and apps that connect directly to cloud storage give you even more options. The best thing to do is to test which combination is best for you in the long run. If you use SharePoint Online in Microsoft 365 and

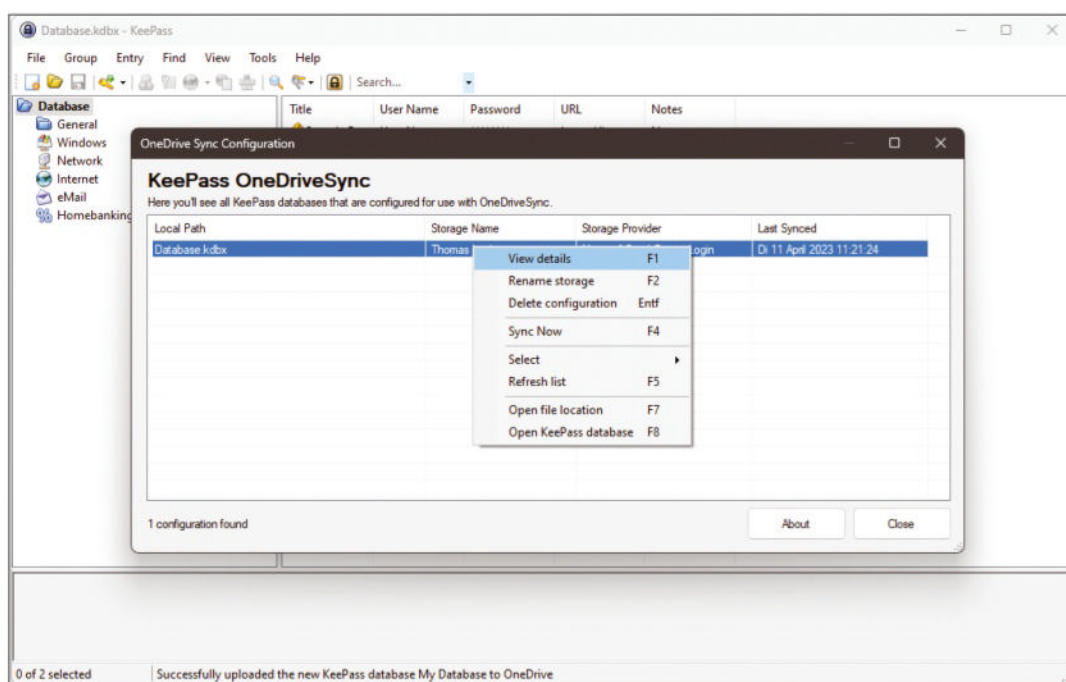


Figure 4: The KeePass OneDriveSync plugin can synchronize databases directly with OneDrive. Several other plugins can do this, as well.

synchronized libraries for data exchange or document storage, you can store the KeePass file directly in the SharePoint library. This method ensures that the latest version of the database file is always available online without additional synchronization because the SharePoint Online library is automatically kept in sync with the local computer by the OneDrive client. The OneDrive app lets you access the libraries and download files in this path, too – and this includes the KeePass file.

For example, in iOS or iPadOS, tap the three dots for the KeePass database in the OneDrive client to access the *Open in Another App* menu. At this point, you can select the KeePass app you have in place and open the file. However, be aware that access in most cases is not online; instead, the cloud storage – OneDrive in this case – downloads the database to the smartphone or tablet. To have the latest version, you need to repeat the process whenever you want to update the database on your smartphone or tablet; you can work with the local file until then.

Of course, if you want to add entries to the smartphone or tablet or change existing entries, you again need to make sure you sync the database file back to your cloud storage so that your other devices can also access the information.

KeePass on macOS

KeePass also runs on macOS, and various apps can handle KeePass databases. One well-known example is KeePassX [3]. If you install the tool on a Mac, you can use your database on the go with a MacBook by using synchronization with cloud storage, just as you would for access by iPhone and Android. Alternatively, you can use MacPass [4]. Both apps

handle KeePass databases and are similar in terms of usage.

After downloading KeePassX, first drag the program to the Application folder, then open the app. If the tool does not start because of security settings, give it permission to start in *System Preference | Security & Privacy | General* in the Apple menu. Create a new database for KeePassX by selecting *File | New Database*. If you already use KeePass on Windows, you can open the database directly at this point. After entering the database password, the tool opens the file, just like KeePass on a PC. You can also assign your passwords to groups to help you find them faster. The *Root* context menu is used to create new password groups, and you can select *Icon* to assign a separate icon for each group in KeePassX.

The context menu in the main KeePassX window is where you add new entries; just select *Add New Entry*. If you store the database in a location that you sync with iCloud, the database will be available on all Apple devices in the account. However, you can also work with OneDrive in macOS and synchronize SharePoint online libraries to open KeePass databases.

Setting up Network Access

KeePass can also be used by multiple users on a network. Of course, you need to share the password file (e.g., via a network drive or a synchronized library in SharePoint Online). In this case, several users can read the data in the file, but only one can write to the file at any given time. KeePass is not network capable itself, but with a little effort it can certainly be used in a team. On the other hand, you could opt for a KeePass server. However, KeePass servers are not available for free and rarely support mobile access.

One example is the Pleasant Password Server [5]. The server gives you a multiuser environment for KeePass and apps for iOS and Android. The server can be used free of charge for up to 30 days.

Each employee can also use their own password file and store team data in the shared access file in parallel. You can use the lock icon in KeePass to block access for other users, if necessary. When this option is enabled, KeePass is minimized and access to the data is not possible again until the password is entered again.

Conclusions

The use of KeePass will undoubtedly require some training if you want to use the tool's database on multiple devices. Because all the data is stored in a single, secure database file, though, you only need a solution for transferring the data to your devices. It makes sense to synchronize with cloud storage. However, password servers can also be used, and synchronization with private cloud environments is no problem. ■

Info

[1] KeePass: [https://keepass.info]

[2] KeePass plugins: [https://keepass.info/plugins.html]

[3] KeePassX: [https://keepassxc.org/download/#macos]

[4] MacPass: [https://github.com/MacPass/MacPass/releases]

[5] Pleasant Password Server: [https://pleasantpasswords.com]

The Author

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

A new approach to more attractive histograms in Prometheus

Off the Chart

Histograms are a proven means of displaying latencies in Prometheus, but until now, they have had various restrictions. Native histograms now provide a remedy. By Julien Pivotto

Prometheus histograms provide a method for displaying the distribution of continuous values. They provide information about the range and shape of the data and are often used to calculate percentiles. To do this, data is divided into 100 distribution areas. The x th percentile is then the value below which x percent of the observations fall.

The classical histogram metric divides a range of values into small sections ("buckets" in Prometheus) and counts the number of observations per area. In classical histograms, you first need to define these areas. Each range is represented by its upper limit. A range of 5s contains the number of all observations with a value less than or equal to five seconds. Besides the ranges, two other values are interesting: the sum total of all observations and the number of observations.

To illustrate, I'll look at a practical example wherein a histogram is defined with three buckets: 1s, 2.5s, and 5s. Two queries, one of which lasts two

seconds and the other four seconds, are then observed. The first observation lies within the 2.5s, 5s, and +Inf ranges, and the second lies within the 5s and +Inf ranges (Figure 1). From

this data, a histogram of HTTP request duration in seconds can be created in Prometheus (Listing 1).

Computing ranges is cumulative. Too many buckets can lead to

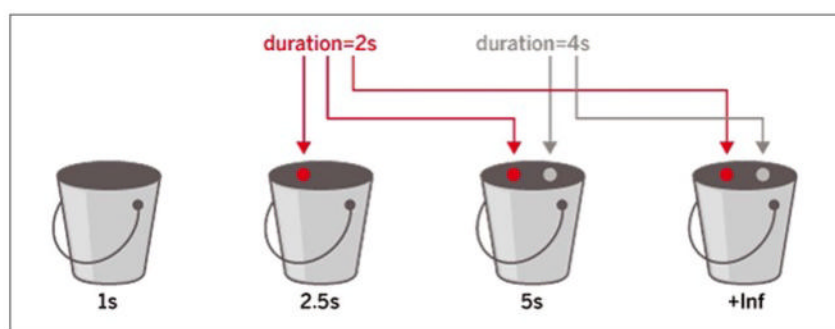


Figure 1: The two-second request is assigned to the 2.5s and 5s ranges, and the four-second request is assigned to the 5s and +Inf ranges.

Listing 1: HTTP Request Duration

```
# HELP http_request_duration_seconds Histogram of latencies for HTTP requests in seconds.
# TYPE http_request_duration_seconds histogram
http_request_duration_seconds_bucket{le="1"} 0
http_request_duration_seconds_bucket{le="2.5"} 1
http_request_duration_seconds_bucket{le="5"} 2
http_request_duration_seconds_bucket{le="+Inf"} 2
http_request_duration_seconds_sum 6
http_request_duration_seconds_count 2
```

performance problems. In such a case, you can delete some ranges with `metric_relabel_configs` while Prometheus is still collecting the values.

Vulnerabilities

Defining range limits is a critical step in classical histograms. The number and size of buckets affect the accuracy of the quantile estimates that can be generated with the histogram. “Quantile” is the generic term for “percentile”; percentiles are the quantiles 0.01 to 0.99 in increments of 0.01.

Ranges that are broken down too finely result in a large number of time series and high memory and disk space use. On the other hand, too coarse a division leads to inaccurate quantile estimates. Finding the balance between the number and size of the domains requires both domain-specific expertise and a good understanding of statistics. In some cases, you might make assumptions that turn out to be unrealistic once your application goes live. Because histograms with different range layouts cannot be merged, changing range boundaries is by no means a trivial task.

The new native histograms make it easier to decide on a range distribution by using a dynamic and adaptive layout. They give you a more accurate representation of the data by independently adjusting the size and number of areas according to the distribution of the data [1].

Another problem with classic histograms is their size and effect on performance. Each area includes a time series that needs indexing and consumes memory; empty areas are always displayed. Additionally, partitioning histograms with multiple labels is ineffective because each label incurs the same amount of overhead, regardless of the number of areas used. For this reason, useful labels such as “error code” are often left out of histograms because each error code value could lead to dozens of new time series.

Native Histograms

Native histograms use an exponential bucket layout with a sparse representation of the data. Empty areas are dropped, reducing memory and disk space overhead. A regular logarithmic range layout with high resolution is used. Also, this type of histogram uses simple ranges instead of cumulative ranges. The application that outputs the histograms determines the schema on the basis of initial hints. The traditional Prometheus time series database does not allow for a sparse display. A classic histogram typically uses two time series for the number and sum total of values, plus an additional time series per bucket, including the empty areas. As a result, many areas very quickly generate a high level of overhead, which is why the new native histograms are stored differently [2]. The Prometheus time series database stores histograms in a structure that contains all sections and the sum total and count together. The application considers these structures to be unique (“native”) objects rather than independent time series (Figure 2). In this way, the time series overhead is incurred only once.

Reused

For many years, Prometheus used a language- and platform-independent mechanism for serializing data in the form of protocol buffers, known as Protobuf, that relies on a binary format instead of a human-readable text format. Over the years, the text format became the norm, but the complexity of the histograms is

bringing Protobuf back into play, because it allows for a very compact representation and makes managing (de)serializing easier. If you enable this option, Prometheus saves all the metrics with Protobuf – if the target supports this format. Metrics can still be collected optionally with tools such as Curl, but doing so forces a return to text format, precluding the use of native histograms.

Switchover

To use native histograms in Prometheus, you need to enable them with the

```
--enable-feature=native-histograms
```

flag. As a result, data collection on the target is done in the Protobuf format. You will want to use the latest available version of Prometheus, because this feature is considered experimental and is still subject to ongoing change. In Go, the *client_golang* v0.14 library supports native histograms, for which you

Listing 2: Mod for Native Histograms

```
requestDuration: prometheus.NewHistogramVec(
    prometheus.HistogramOpts{
        Name:    "prometheus_http_request_duration_seconds",
        Help:    "Histogram of latencies for HTTP requests.",
        Buckets: []float64{.1, .2, .4, 1, 3, 8, 20, 60, 120},
        NativeHistogramBucketFactor: 1.1,
        NativeHistogramMaxBucketNumber: 150,
    },
    []string{"handler"},
)
```

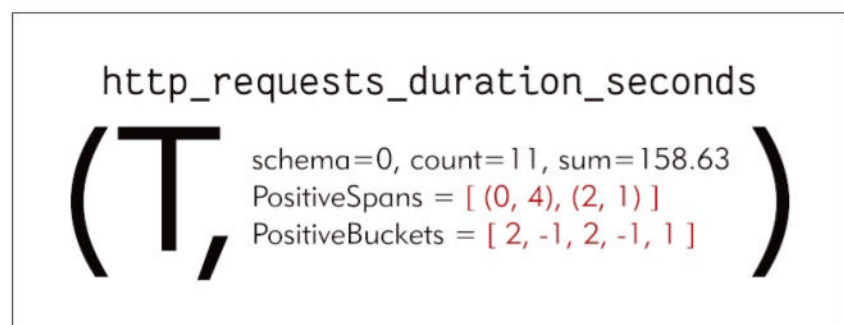


Figure 2: A native histogram is stored with all areas as a single object. All non-empty areas (and only these) belong to a time series

need to modify the code as shown in [Listing 2](#).

In slightly simplified terms, native histograms use exponentially staggered buckets to cover the entire `float64` range of values. The width of the ranges increases by a constant factor that can be adjusted to balance overhead and accuracy. A factor of 1.1 (each successive range is 10% wider than the previous one) provides a good compromise and results in eight ranges per power of two (e.g., between 1 and 2, 2 and 4, 4 and 8, etc.).

You can specify the maximum number of buckets with the

```
NativeHistogramMaxBucketNumber
```

option. Without this setting, the number of areas could grow uncontrollably and cause high memory usage. Limiting this option to a defined number allows the native histogram to recalculate its ranges and aggregate some of them, if needed, to reduce memory usage.

Results

Figure 3 shows a comparison for computing the 95th percentile of a specific query for identical data. At each point on the *x* axis, the corresponding *y* value represents the threshold below which 95 percent of the requests fell at that point. On the left you see the computations with a legacy histogram, and on the right with the native variant.

The comparison shows the native histograms provide a far more accurate estimate of the percentile value. In contrast, legacy histograms can misleadingly suggest the values are identical before and after a peak. In fact, the percentile value in the example before the peak is about 350ms, and about 500ms after the peak. This difference of 150ms is not represented by legacy histograms.

Conclusions

The Prometheus data model has experienced a major change with the addition of native histograms [\[3\]](#). This new feature offers a more accurate

representation of the data and allows for superior computation of percentiles compared with legacy histograms. The move to Protobuf to collect metrics is a significant upgrade. ■

Info

- [1] Native histograms in Prometheus: [\[https://docs.google.com/document/d/1cLNv3aufPZb3fNfaJgdaRBZsInZKKIH09E6HinJVbpM/edit\]](https://docs.google.com/document/d/1cLNv3aufPZb3fNfaJgdaRBZsInZKKIH09E6HinJVbpM/edit)
- [2] Native histograms in Prometheus (keynote video): [\[https://promcon.io/2022-munich/talks/native-histograms-in-prometheus/\]](https://promcon.io/2022-munich/talks/native-histograms-in-prometheus/)
- [3] PromQL for native histograms (keynote video): [\[https://promcon.io/2022-munich/talks/promql-for-native-histograms/\]](https://promcon.io/2022-munich/talks/promql-for-native-histograms/)

The Author

Julien Pivotto has been instrumental in the development and evolution of Prometheus as the maintainer. He is one of the founders of O11y ([\[https://o11y.eu\]](https://o11y.eu)), a company that provides premium support for various open source monitoring tools such as Prometheus, Thanos, and Grafana.

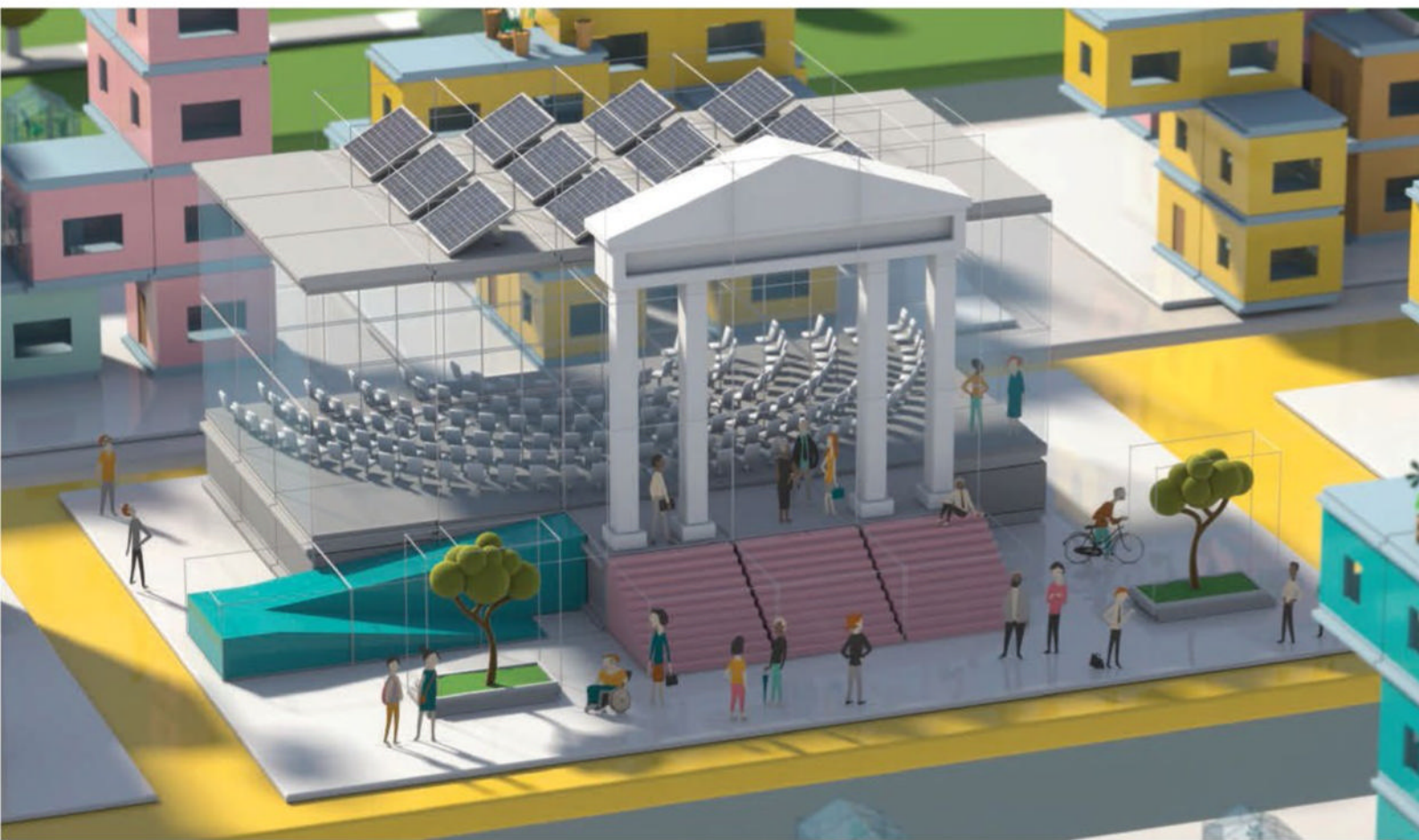


Figure 3: A comparison computing the 95th percentile of a specific query over time with legacy (left) and native (right) histograms.

© PromCon EU 2022 [3]

Public Money

Public Code




Modernising Public Infrastructure with Free Software



Free Software Foundation Europe

Learn More: <https://publiccode.eu/>



Manage projects in SMEs with OpenProject

Best-Laid Plans

OpenProject supplies sensible, comprehensive project management for SMEs with few financial inputs.

By Martin Loschwitz

Many project management tools are complex and require large teams. Still, it is unwise for IT departments in small and medium-sized enterprises (SMEs) to do entirely without planning because the benefits massively outweigh the overhead. I look at project management from the theoretical and practical side and, after a brief introduction to the different types of projects, investigate how OpenProject can help IT managers.

Project Management

An entire industry focusing on project management has established itself in the enterprise sector. Scrum, Kanban, and many other approaches are promoted and sold along with tools and training. Major corporations have project management departments that dispatch their agile coaches to corporate departments on a needs-driven basis.

However, project management is largely dead in the water in smaller

organizations. SMEs are usually not interesting enough for the large providers of agile training and certification because not enough revenue can be generated. Therefore, in many SMEs, project management is largely unexplored territory or, at best, restricted to the framework of what the typically small number of admins can handle. The idea that project management is not needed is a dangerous fallacy because many of the tasks involved in maintaining and operating servers in large corporations also occur on a smaller scale in SMEs. These tasks must be completed just as diligently as in larger enterprises to avoid problems arising. SMEs, however, have significantly fewer human resources available to perform these tasks. As a rule, small businesses will need even better project management than large corporations; yet the SME is the place where project management is often in a deplorable state. This state of affairs does not have to be the default. Tools and instruments

on the market make it possible to plan and keep track of projects, especially in the open source world.

Central Aspects

Project management is a bit like the cloud: Everyone who uses the word associates it with certain procedures, requirements, and processes; however, when two people talk about project management, it is quite likely they are imagining different things. Therefore, it makes sense to look at the fundamentals and several dimensions of project management before moving on to the practical implementation.

The first question is: What is a project? Administrators are confronted with different types of input in everyday life, each of which requires a different approach. For example, a help desk cannot be outsourced to a dedicated team in the vast majority of SMEs because a dedicated team simply does not exist. Instead, the IT team, which often

Photo by Pedro Miranda on Unsplash

consists of just one or two people, has to be the help desk. The problem with a classic help desk is that it can hardly be planned or spread in a meaningful way. If a user has a problem with the IT infrastructure, they usually expect immediate help because they otherwise simply cannot work. This kind of work has nothing to do with work on a project, but admins still have to keep it in mind, because it consumes a considerable amount of their time.

Work on the infrastructure can be planned. At least in principle, it is closer to the classic definition of a project. Installing new network storage, for example, can be such a case, and it can be broken down into quite a few smaller units. From planning, to requests for quotes, procurement, physical installation in the rack, and logical installation through commissioning, the process is delicate and usually has to be completed in stages. In this case, classic project management on the waterfall principle shows its greatest efficiency, because infrastructure work is difficult or impossible to implement in an agile manner. Scrum and Kanban also envisage a development process divided into many small parts, but these parts are independent of each other over time. The question of whether feature A or B is implemented first depends on financial and product issues rather than physical necessity. In the case

of infrastructure work, on the other hand, the individual steps need to be handled in sequence. A server that is not installed in the rack because it was delivered late cannot be given an operating system.

Agile project management for software development, however, is practically nonexistent in many companies because of the lack of in-house developers. Much depends on whether the SME itself is involved in software development as a field of activity. A small company that develops its own programs will usually take care of managing its internal IT as a side effect. However, the persons doing the work then have two jobs and would do well to demarcate those jobs. Typical agile approaches and tools can certainly be used to develop a company's products – also within the framework of a larger team. However, in the Internet help desk role, it makes sense for IT employees to use the appropriate tools for completing internal projects, rather than tools that they may already be familiar with from their other role. On the other hand, people who operate the infrastructure in a law firm or carpentry shop will only rarely be confronted with this problem. No software development in the true sense takes place in these places; consequently, there is practically no need for agile methods.

The Right Tool at the Right Time

For SMEs in particular, choosing the right tool is of critical relevance. The bad news, however, is that no one tool covers multiple areas especially because the tasks at hand all differ. Help desk tasks in SMEs should always be managed by software designed for this purpose. In this context, the smaller the company, the more sense it makes to use a cloud-based help desk. A number of ticket and incident management systems are on the open source market (e.g., OTRS or Request Tracker), but they are typically optimized for use in corporate groups and are therefore extremely feature rich. If you only need a tool to communicate with users in a structured way, OTRS will overwhelm you with its feature set. Tools like Zendesk (Figure 1) or Freshdesk are the way to go, even if they cost money.

The choice of tools for agile project management in an organization should ideally be left to the implementing teams. Again, specialization is the be-all and end-all. If teams develop software in an agile manner, Scrum or Kanban can be useful approaches, but the decision should be up to the developers themselves. For traditional in-house IT in SMEs, that leaves only one type of project that occurs on a regular basis: infrastructure work on the waterfall principle. The good news is that admins who otherwise largely fight a losing battle without any help can seriously simplify their lives with a basic but functional project management tool. However, quite a few applications on the market vie for your favor – even if you restrict the field to tools with an open source background. That said, the top dog in this segment is OpenProject [1].

OpenProject for SMEs

OpenProject comes in several variants. Although the tool is open source software, the manufacturer offers commercially licensed products in the form of the Professional and

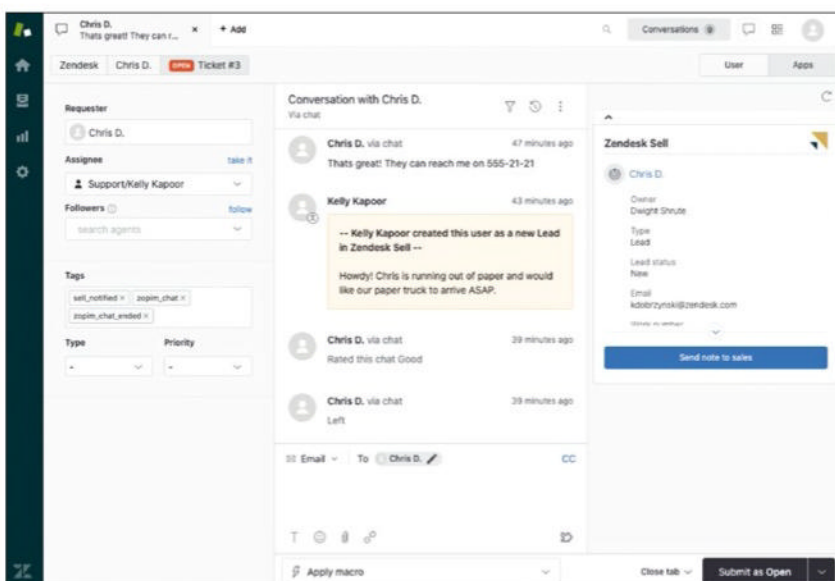


Figure 1: The right tool for managing support tickets is help desk software like Zendesk.

Enterprise variants. The Community edition is open for everyone to use without license fees. The functionality the Community edition offers is absolutely fine for the vast majority of SMEs and their projects. The main advantage of the software is that it does not confront you with too many functions not needed in everyday life. Another advantage is that the tool is web based; at its core it is a server application. The end effect is that OpenProject can be operated from any computer in a browser and that everyone accessing the tool sees the same status for every project. Other tools sometimes come as client applications, but then need a method in the background to keep their local datasets in sync. These factors are completely eliminated with OpenProject. The OpenProject developers also offer a prefabricated Docker container. In principle, this can be started on any server with a working Docker runtime environment. The following example assumes you have a fresh Ubuntu 22.04, which can be virtualized, of course. Installing the Community edition of the Docker environment is described online [2], so I will not go into this aspect of the setup in detail.

Getting Started

Once a system with a working Docker environment is in place, you still need to complete a little preliminary work because OpenProject accesses a PostgreSQL database in the background to store its project data. The database itself is implicitly part of the container that also delivers OpenProject, but for the data you store in OpenProject to persist, the Docker container needs two volumes where it can permanently store the PostgreSQL data. You can create these on a standard Ubuntu and launch OpenProject with the commands in [Listing 1](#).

Replace the value of `OPENPROJECT_HOST_NAME` with the hostname under which you will be accessing OpenProject. Very importantly, this hostname needs to be resolved locally by DNS. You do not need publicly resolvable hostnames, as long as the configuration on the local DNS server is correct. You also need to replace the password defined by `OPENPROJECT_SECRET_KEY_BASE` with a genuine password. Although OpenProject will run now, that does not mean it is ready for use. Once again, SSL rears its head. The developers assume that OpenProject is not aware of SSL itself. Instead, the application expects to reside downstream of a load balancer or a reverse proxy that takes care of SSL termination. If you already operate a terminating proxy or load balancer locally, simply store the appropriate configuration for the OpenProject hostname on it and make sure the HTTP X-Forwarded-Proto flag is forwarded to OpenProject. If a proxy is not yet available locally, online documentation [3] describes how an HAProxy with the corresponding configuration can be implemented with a Docker container. The HAProxy should always be configured to point to port 8080 on the host on which OpenProject is running.

Creating or Importing Users

The OpenProject instance that has just been rolled out is still empty because you have not created any projects to date. After deployment, you will want to take a closer look at the user interface. After the initial installation, the login name and password are *admin*, and your first step will be to create some users. How this works depends on your local setup. The easiest way to create accounts for your users is to set them up on the OpenProject admin console.

Many SMEs today have LDAP or Active Directory user directories, so if a directory of this kind exists, you will want to connect it to OpenProject; otherwise, you are forced to maintain user data in multiple, independent locations, which makes no sense from either a convenience or compliance perspective.

OpenProject can also connect to an existing user directory from its administration console. What the community edition does not give you, however, is the ability to import from the directory service. You might need to maintain these groups at the OpenProject level as required.

Starting Your First Project

You will find a great deal of relevant information on the start page. In addition to a welcome message is the Projects section that lists all projects to which a user has access. OpenProject defines a tree-like hierarchy for projects and sub-projects. In SMEs, however, this functionality is often not fully utilized. A top-level project named “Internal IT” and sub-projects such as “Installation of new NAS” or “Replacement of database server” would be conceivable and useful here. You are completely free to choose the structure.

To add a project, just click on the + *Project* entry in the drop-down menu at top left or on the button of the same name on the start page. Once the desired structure has been created, a click on the selection menu at top left or the *Projects* cube on the start page is the next step.

You are almost done now, and OpenProject shows you – for the first time – the typical overview page for the project you just created; of course, this page is still empty. However, on the left you will now see several menu items that let you to create and manage tasks.

Listing 1: Preliminary Work

```
mkdir -p /var/lib/openproject/{pgdata,assets}
sudo mkdir -p /var/lib/openproject/{pgdata,assets}
docker run -d -p 8080:80 --name openproject -e OPENPROJECT_HOST_NAME=openproject.example.com -e OPENPROJECT_SECRET_KEY_BASE=secret -v /var/lib/openproject/pgdata:/var/openproject/pgdata -v /var/lib/openproject/assets:/var/openproject/assets openproject/community:12
```

The Correct Structure

As is so often the case with project management, a little thinking up front avoids major headaches further down the road. When you create a project and its associated tasks in OpenProject, think carefully about the level of granularity you need. For example, is it fine to have a *server installation in rack* task because you are the only person performing it anyway. On the other hand, are several people involved in a process, such as the purchasing department that has to order cables and make sure they are delivered?

All work steps of a project should be mapped in OpenProject as fully as possible to reduce the risk of forgetting small tasks outside your own area of responsibility (e.g., the server is in the right rack, but the cables you need to install it are missing).

In essence, working with OpenProject differs very little from working with other tools of its kind. Each individual task comes with its own description, due date, and possible dependencies on other tasks. The more carefully you maintain the individual

tasks, the more meaningfully you can work with the software. The tool's built-in Gantt function is particularly useful for infrastructure projects that can only be planned with a waterfall approach. A Gantt table represents all pending tasks along a timeline. Because Gantt tables can also be exported from OpenProject, a finished project plan can become your printed project wallpaper that visualizes the progress of all the tasks in a friendly analog way (Figure 2).

One major advantage of OpenProject is that it is easy for newcomers to get started. Most of the descriptions and procedures are logical and easy to follow. After a short learning curve, during which the OpenProject documentation [4] will become your best friend, you should be familiar with working with the software. From now on, this software will serve you as a powerful project manager – even for small projects – and can be used to apply structure to what might otherwise be chaos.

Conclusions

Before you start using any tool for project management, you also need

to categorize the tasks at hand. More so than virtually anywhere else, having the right tool for a job is key to efficient work in SMEs. The bulk of administrative tasks consists of traditional infrastructure operations, and waterfall-based approaches have been proven to work best. OpenProject gives you a free tool for precisely this task, offering a manageable function set that does not frighten off newcomers with feature overkill. ■

Info

- [1] OpenProject: [\[https://www.openproject.org\]](https://www.openproject.org)
- [2] Docker CE installation: [\[https://docs.docker.com/engine/install/ubuntu/\]](https://docs.docker.com/engine/install/ubuntu/)
- [3] HAProxy as an SSL terminator: [\[https://serverok.in/enable-ssl-in-haproxy-docker-container/\]](https://serverok.in/enable-ssl-in-haproxy-docker-container/)
- [4] OpenProject documentation: [\[https://www.openproject.org/docs/\]](https://www.openproject.org/docs/)

The Author

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

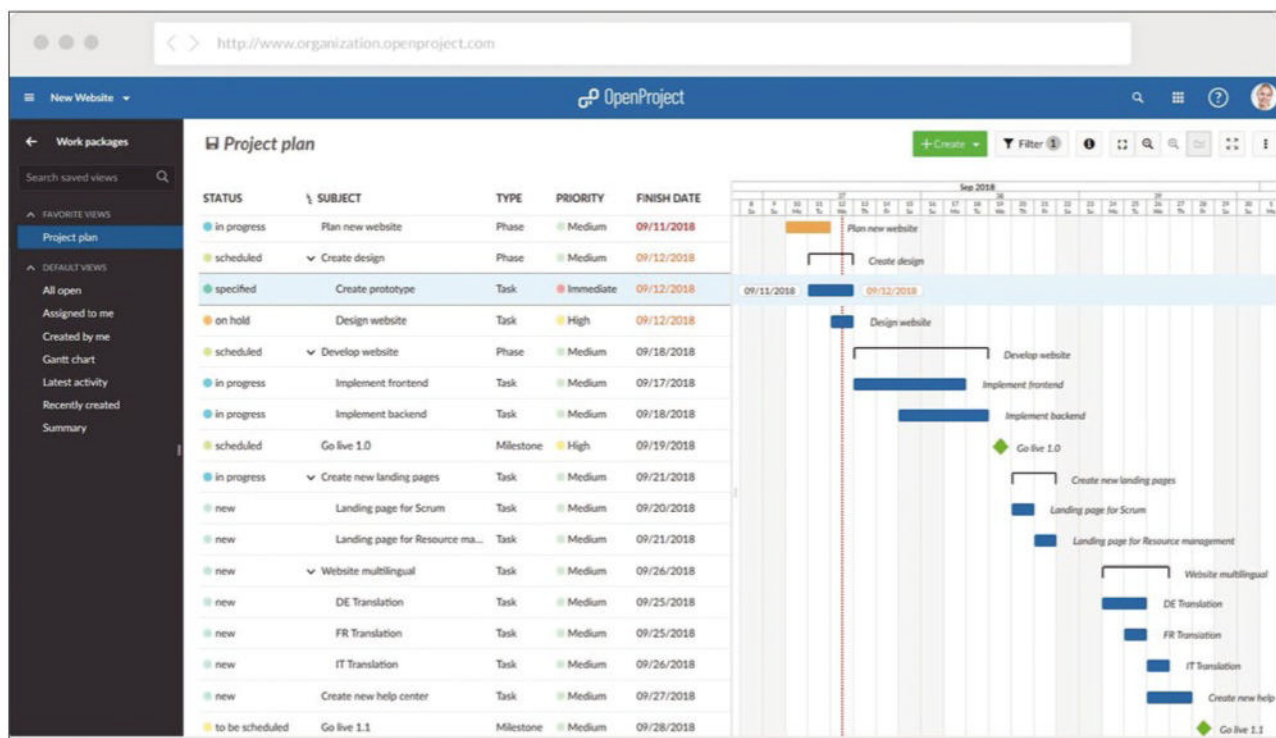


Figure 2: Gantt charts provide a quick visual overview of the status of current tasks in classic waterfall projects.

Open source monitoring with Zabbix

A Clear View

Zabbix provides comprehensive, highly configurable, but easy-to-use system monitoring. By Christian Anton

Some particular strengths of the Zabbix monitoring solution are constant development according to user requirements, wide number of use cases, and wide range of integration options. Whether monitoring classic servers or arbitrary networks, operational technologies (OTs) and Internet of Things (IoT) devices, or modern containers and cloud-based infrastructures, Zabbix plays to its strengths. These strengths include multiple data collection methods and easy-to-define but extremely flexible problem detection and alerting. Additionally, Zabbix has attractive visualizations, a graphical configuration tool, and web and integrated business services monitoring with service level agreement (SLA) reports. The manufacturer of Zabbix distributes its software fully under the GPLv2. Every user, from private to large corporations, is given access to the same product; no enterprise functions are hiding behind a paywall. The Zabbix business model is based on support and consultancy services and training and has a worldwide partner network. Major release updates for the monitoring tool arrive every six months.

Software packages are available from the in-house repositories of all well-known Linux distributions, such as Red Hat Enterprise Linux, CentOS Stream, Debian, and Oracle. Besides Linux, Zabbix also supports other Unix-style systems, such as HP-UX, Solaris, and even macOS. However, the manufacturer recommends Linux. Official Raspberry Pi and container image packages are also available.

Basic System

The core of a Zabbix monitoring system is the Zabbix server, the central control unit that collects, (pre)processes, evaluates, and stores the data in a relational database. The Zabbix web interface is a PHP application that accesses the Zabbix database to visualize the data collected by the server. It provides an interface for the administrator, but a Zabbix API also provides an interface with arbitrary integration and automation tools. Zabbix configuration takes place in the web interface. When it comes to databases, Zabbix falls back on tried and trusted resources and supports MySQL, its descendants MariaDB and Percona,

the popular open source PostgreSQL, and the commercial Oracle product. In addition to the core components, two variants of the Zabbix agent, a web services component, currently have the sole task of emailing regular reports. Finally, the Zabbix proxy enables distributed monitoring with decentralized data collectors.

Installation

A basic installation is a quick experience. After installing all the Zabbix software components and the database management system with the package manager of your choice, you initialize it with the database schema required for Zabbix. Doing so generates the tables needed for configuration, historical data, some basic settings, and an initial host. In the next step, you need to configure the Zabbix server with the database access credentials and install the required software components for operating a PHP-based web application. The software packages from Zabbix make commissioning very easy. The Zabbix website provides clear step-by-step instructions for the most widely used Linux distributions. In

Photo by Vincentiu Solomon on Unsplash

principle, all Zabbix components can be run on a single (physical or virtual) Linux server. Nevertheless, Zabbix recommends running the individual components on separate servers in larger production environments. In particular, the database required for Zabbix is best run on a separate server as soon as the number of hosts to be monitored exceeds a certain number, or if operating the monitoring solution is a strategic part of your IT operations. The database management system (DBMS) has to cope with high-volume write access, because Zabbix stores all collected data in various database tables. A mechanism ensures that the data does not expand uncontrollably. In a larger monitoring environment, the Zabbix database can easily grow to several terabytes, depending on the volume of, and the retention time configured for, the collected metrics, events, audit logs, and so on.

As an alternative, Zabbix can be rolled out in containers [1]. Besides official container images for all Zabbix software components, the vendor offers prefabricated Docker Compose manifests that let you quickly and easily launch the solution in Docker.

Helm charts developed by the open source community and a lightweight OpenShift operator provided by Zabbix also are available for operation with Kubernetes and OpenShift. The approach I recommend is the Helm chart [2]. I am involved in this as a maintainer, and my employer uses it in many production environments, some of which are very large. Container-based deployment is a very practical option: For example, you can quickly set up a small, fast Kubernetes cluster on a local machine in minikube, in K3d, or with Docker Desktop or Rancher Desktop. Additionally, you can quickly and easily set up a full Zabbix monitoring stack with:

```
$ helm repo add zabbix-community 2
https://github.com/zabbix-community/2
helm-zabbix
$ helm install 2
-n zabbix 2
--create-namespace zabbix 2
--set zabbix_image_tag: 2
alpine-6.4-latest 2
--set zabbixweb.service.type=NodePort 2
zabbix-community/zabbix
```

This sample code sets up a PostgreSQL database (without persistent

storage, all data is deleted after shutdown), together with the Zabbix components, and provides the port for the Zabbix web interface as a NodePort service. This example does not include an Ingress object or a TLS certificate. Matching documentation can be found on the GitHub page of the Helm chart project.

To point to the service that provides the Zabbix web interface, use

```
$ kubectl -n zabbix get service 2
-1 app.kubernetes.io/instance=2
zabbix-zabbix-web
```

which states the port on which the interface can be accessed. After successfully navigating the installation, the Zabbix web interface welcomes you and invites you to log in. After logging in as *Admin* with password *zabbix*, you are taken to a central dashboard (Figure 1).

Web Interface

The Zabbix web interface is divided into six areas, with some external links to Zabbix documentation and Zabbix's own integration website, where monitoring templates and

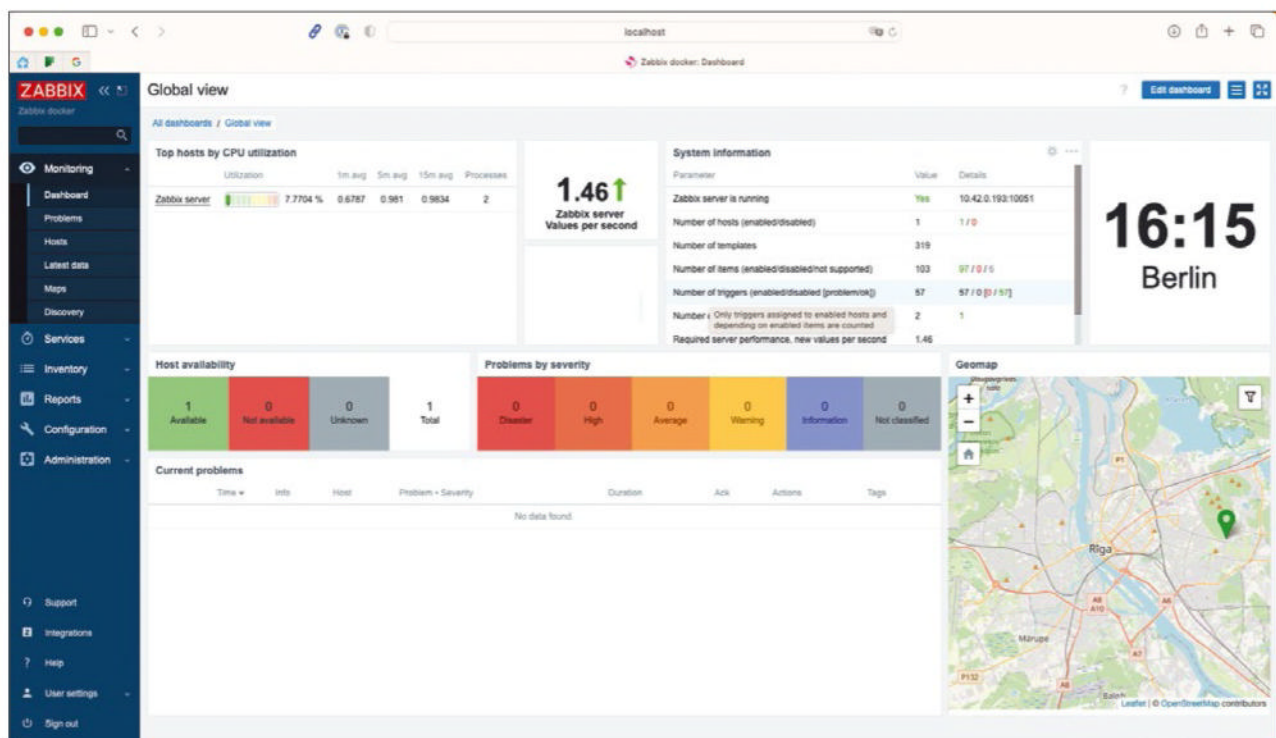


Figure 1: The default Zabbix dashboard with some basic information.

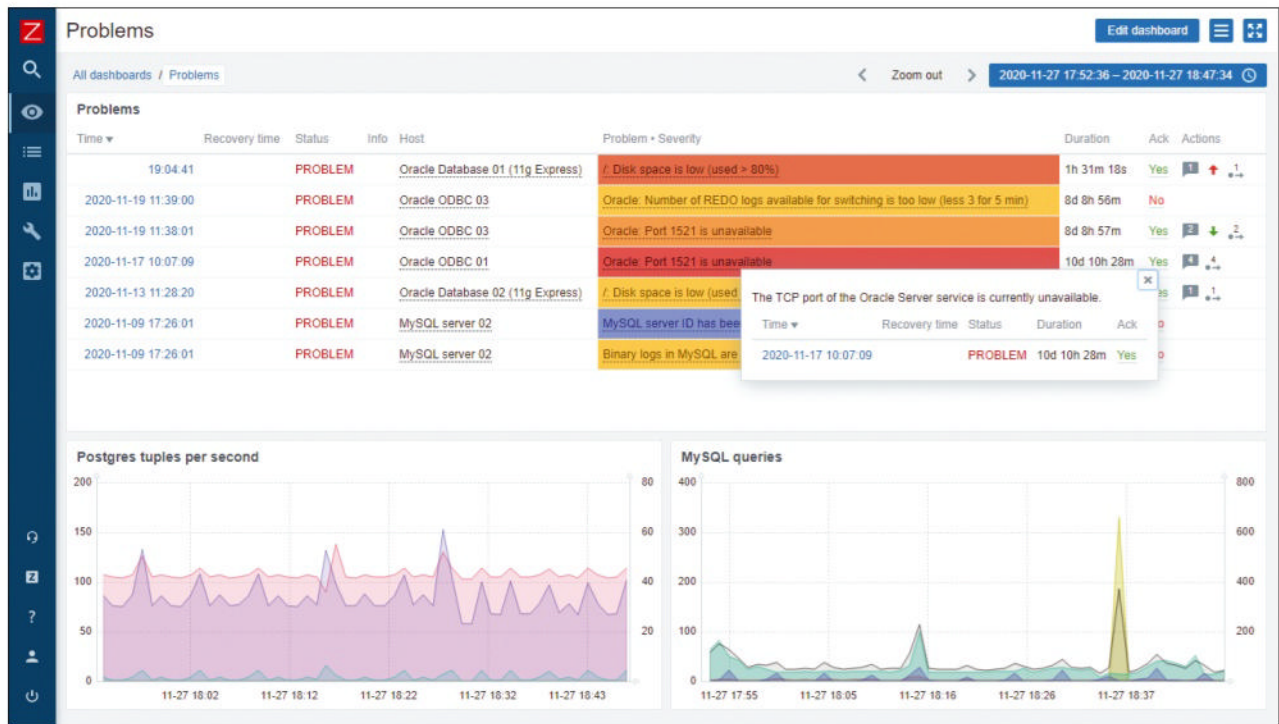


Figure 2: The Zabbix dashboard shows historical data and points to problems.

other useful things can be found. The *Monitoring*, *Services*, and *Inventory* sections let you view the monitoring data (Figure 2), work on any issues that arise (acknowledgment, ticket generation, etc.), and analyze system and application states retrospectively with graphs and other historical views (Figure 3). The services in Zabbix 6.0 are completely new, with powerful features designed for monitoring and alerting with more than 100,000 business services. It is used to derive the status of operational processes from acquired metrics data. Other new features include an SLA reporting function and the ability to trigger alerts according to the states of business services. Scheduled reports (*Reports* | *Scheduled reports*) lets you define dashboard-based reports and mail them at regular intervals. Scheduled reports can also be used in conjunction with the *SLA report* dashboard widget to report SLAs to application owners on a regular basis (e.g., weekly). The dashboards have also seen many changes in recent Zabbix versions (Figure 4). You can now build modern dashboards that look as attractive as those created with dedicated

dashboard tools like Grafana. The foundation for this feature was laid in Zabbix 3.4: The current version of the new dashboard framework is based

on freely placeable widgets. Over the past few years, these widgets have been continuously developed to include service views, vector-based



Figure 3: Graphs are used for retrospective problem analysis.



Figure 4: An example of a modern Zabbix dashboard with the Dark theme.

and dynamic graph views, geographic maps, network maps, and far more. Zabbix also has much to offer under the hood. Extensive permissions configurations and the role strategy integrated into Zabbix make it possible to display exactly the monitoring objects required for the role in question (Figure 5). Also, areas of the web interface can be specifically released for individual users or user groups, making it possible to allow selected users to view large parts of the infrastructure, but only on the basis of individual dashboards that you can configure fully.

Also important for enterprise use is the ability to authenticate to the Zabbix system by one or multiple Lightweight Directory Access Protocol (LDAP) and Active Directory (AD) servers. Zabbix can be connected to a modern Security Assertion Markup Language (SAML)-based identity management system (e.g., to the open source Keycloak solution) or to one of the numerous cloud-based providers (e.g., Okta, OneLogin, Microsoft Azure AD).

At the time of writing, a super administrator still had to configure all the users and their assignments manually to roles and permission groups within Zabbix. Since then, Zabbix

6.4 has been released with just-in-time (JIT) user provisioning [3]. JIT makes it possible to use LDAP or SAML attributes to assign appropriate group and role memberships to users as soon as they log in and thus automatically enforce an authorization concept.

All system monitoring configuration is handled by the Zabbix web GUI, where you can look forward to a sophisticated template-based system.

To automate workflows and integrate with third-party systems, a JSON remote procedure call (RPC) API provides all of Zabbix's functionality [4]. Python and the Ansible and Salt Stack automation frameworks (among others) have modules for this programming interface; integration with ticket tools is also supported. Zabbix also benefits from the completely free availability of

more than 300 monitoring templates for a variety of popular operating systems, network devices, applications, and cloud services (Figure 6). The community also provides a ready-made and easily customizable monitoring template for virtually every use case.

Functionality

Zabbix takes a metrics-based approach. Data is initially collected either by the Zabbix server or by a



Figure 5: Zabbix offers a whole range of user authentication options. © Zabbix

Zabbix proxy and stored centrally in the server database for analysis. The metrics – “items” in Zabbix-speak – are initially stateless and represent the raw data for visualizations and problem detection. Logical expressions, known as trigger expressions, are used to identify problems. The server evaluates them for each new record that reaches Zabbix. A variety of item types are available in Zabbix that implement different data collection methods. The Zabbix agent is certainly the most effective and simplest option for classic operating system and application monitoring.

Agents

The Zabbix agent is available directly from the Zabbix website and package repositories for all major operating systems and architectures. Zabbix 5.0 also added a more modern agent 2, which is only supported for Linux and Windows. It is functionally compatible with the traditional Zabbix agent written in C but has been implemented in Go

to provide advanced features, such as native monitoring of MySQL, PostgreSQL, and Oracle databases; monitoring of TLS certificates; a subscription to MQTT (a machine to machine network protocol) topics; and much more. Especially exciting is that it can be extended with Go modules, which offer features such as persistent connections, your own program code running permanently with the agent, or the ability to implement complex business logic directly in the agent. Both Zabbix agents can transmit data actively and passively to the Zabbix server or a Zabbix proxy, which enables flexible deployment scenarios adapted to the network situation and performance requirements. Additionally, agent 2 buffers the acquired metrics data in a local SQLite database on demand in case the proxy or server becomes unreachable. This feature is particularly useful for IoT applications. The native feature set of Zabbix agents [5] includes monitoring CPUs, memory, and disks, as well as advanced features such as Windows management

instrumentation (WMI) queries, inventory queries, file and directory functions, and more.

Other Data Sources

In addition to the many ways to collect data about agents, simple communications connections (e.g., ICMP and TCP/UDP) are checked, and metrics such as response times are collected. Simple Network Management Protocol (SNMP) items let you query values by SNMP and field SNMP traps. These items support all SNMP protocols, security levels, encryption, and password hashing.

Also, you can execute SSH/Telnet commands and store their results, run queries by any Microsoft Open Database Connectivity (ODBC) database, read data from Java-based applications with the Java management extension (JMX), and collect sensor data from server hardware over the Intelligent Platform Management Interface (IPMI). Internal items reflect metrics that map the state and performance of the Zabbix monitoring engine itself and are used for

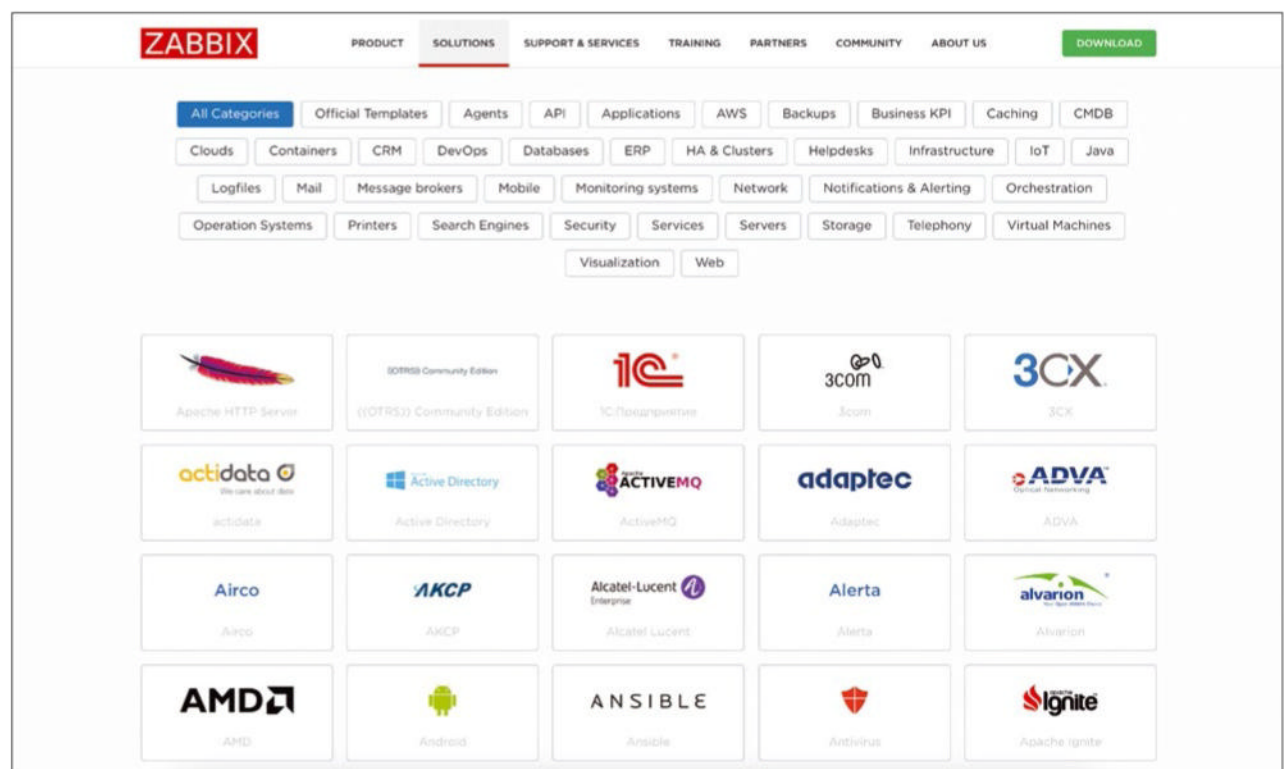


Figure 6: The Zabbix website illustrates the many possibilities of integration with third-party software.

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
delivered every year



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



@adminmagazine



@adminmag



ADMIN magazine



@adminmagazine

self-monitoring of both the Zabbix server and the Zabbix proxies.

In recent years, Zabbix has invested a great deal of development in advanced data acquisition designed to do justice to the self-advertised all-in-one approach of the monitoring solution and to implement special monitoring requirements without the use of external scripts, as far as possible. In the meantime, you have extended item types, such as the HTTP item, which queries data from status web pages or the API endpoints of applications, or the script item, which enables virtually arbitrary function definitions with an integrated JavaScript interpreter. For use cases that go beyond the supplied item types, the functionality of Zabbix itself, and the Zabbix agents, can be extended in many ways. The options range from including simple scripts, through extensions of the C-based monitoring core of the Zabbix server and proxies, to

the previously mentioned Go plugins in agent 2.

Calculated items are used for aggregations and derivations from measured values. They have also seen a comprehensive feature update in version 6.0. For example, it is now possible to use tag markers and aggregation functions to perform dynamic computation on arbitrary, flexible numbers of items and save the results as another item. In this way, you can discover, say, how many systems in a certain category have a CPU load that exceeds a predefined threshold over a period of time. In the same way, you can count the total number of active users across all nodes in an application cluster. Another recent important building block that has significantly revolutionized gathering data in Zabbix is the combination of dependent items and preprocessing. A variety of functions are available to preprocess monitoring data, ranging from simple text replacements with search

and replace or with regular expressions, to advanced data extraction with JSONPath and XML Path and validation with flexible problem handling, to custom preprocessing with JavaScript.

For example, values can be extracted from JSON fragments or HTTP queries and processed as metrics for monitoring. The process of extracting multiple values from a single dataset with an HTTP item and distributing them across multiple dependent items now runs smoothly and does not rely on the use of external scripts.

Problem Detection

Just as flexible as the options for data collection are the functions for generating events from the acquired and historicized raw monitoring data, and thus for presenting problems.

The introduction of the new trigger expression syntax in Zabbix 5.4 is certainly the most important change

The screenshot displays the Zabbix configuration interface for defining a trigger. The main form is titled 'Triggers' and shows a list of triggers with columns for 'Trigger', 'Tags', and 'Dependencies'. The selected trigger is 'Some Problem has occurred!'. The form fields include:

- Name:** Some Problem has occurred!
- Event name:** Some Problem has occurred!
- Operational data:**
- Severity:** Not classified, Information, **Warning**, Average, High, Disaster
- Expression:** (with an 'Add' button and an 'Expression constructor' link)
- OK event generation:** Expression, Recovery expression, None
- PROBLEM event generation mode:** Single, Multiple
- OK event closes:** All problems, All problems if tag values match
- Allow manual close:** ☐
- URL:**
- Description:**
- Enabled:** ☒

A modal window titled 'Condition' is open, showing the configuration for a condition:

- Item:** Zabbix server: Available memory (with a 'Select' button)
- Function:** last() - Last (most recent) T value
- Last of (T):** Count
- Time shift:** now-h
- Result:** < 1G

The modal window has 'Insert' and 'Cancel' buttons.

Figure 7: Defining a trigger in the configuration interface.

in recent years. The manufacturer has changed the notation of expressions to a new format that should be easier to interpret and understand. Along with the new syntax came numerous new trigger functions [6], totaling about 100 mathematical, logical, string-based, iterator, and aggregation functions.

The multitude of available trigger functions might seem a bit confusing at first glance; however, with the graphical expression builder (Figure 7) and the *Add* button to the right of the trigger Expression window, it is very easy to create triggers. You simply select the item on the basis of which you want to set a trigger, then choose a suitable function, and parameterize it in line with the short description. Done.

Instead of storing threshold values in a fixed trigger definition, the use of user macros is recommended because they offer an additional customization option and are used to store trigger definitions in templates, which in turn can use multiple hosts (and even other templates). User macros can be assigned a default value, globally or in a template, that can be overwritten host by host. This capability allows individualized triggering per host and use case. A trigger expression could look like:

```
last(/Zabbix server/vm.memory.size[available])<{$MEMORY_LOW}
```

You then need to define the corresponding user macro for this trigger

function as shown in Figure 8. You always need to specify the item referenced in a trigger definition with its item key. The key uniquely identifies each item on a host or in a template and can occur only once in the corresponding context.

Trigger

One big benefit in Zabbix is the metrics history, which is always available for trigger evaluation. You can minimize false positives by evaluating the history instead of individual measured values (i.e., monitoring alerts caused by conditions that occur for a short time but do not represent a malfunction). For example, a simple trigger expression such as

```
max(/<Host>/icmpping,3m)=0
```

could be used to respond only if a host has not been reachable by an ICMP ping for at least three minutes. However, triggers can be much more complex. Combining multiple sub-expressions with advanced trigger functions lets you calculate changes in values over time or determine the direction and continuity of the change. You could even have evaluation functions that are based on machine learning that incorporate long-term data and implement prediction functions.

As an example of an extended trigger expression, I'll look at the trigger anchored in the supplied templates for low space on a filesystem. This

example combines two definable thresholds. One relates to the utilization level as a percentage and the other to a fixed value for the remaining available space. It also has a prediction function:

```
last(
  /Linux filesystems by Zabbix agent/
  vfs.fs.size[{$FSNAME},used])>
{$VFS.FS.PUSED.MAX.WARN}
"{$FSNAME}" and
((last(/Linux filesystems by Zabbix
  agent/vfs.fs.size[{$FSNAME},
  total])-
last(/Linux filesystems by Zabbix
  agent/vfs.fs.size[{$FSNAME},
  used]))
<{$VFS.FS.FREE.MIN.WARN:"{$FSNAME}"
or timeleft(/Linux filesystems by
  Zabbix agent/vfs.fs.size[
  {$FSNAME},used],1h,100)<1d)
```

This trigger only fires when the percent threshold is exceeded but, at the same time, either the value drops below a fixed threshold in megabytes or the filesystem is less than one day away from becoming 100 percent full.

Users can view and edit the problems detected by the triggers on the Problems page of the user interface. You can filter by problem, item, host and template tags, host groups, problem and host names, and monitoring host meta-information. Filters can also be saved to ensure quick access at any time. You have several options for changing how the Problems page (Figure 9) displays. The popular

The screenshot shows the 'Host' configuration window in Zabbix, specifically the 'Macros' tab. It displays a table with columns for 'Macro', 'Value', and 'Description'. A macro named '{\$MEMORY_LOW}' is defined with a value of '1G' and a description 'amount of memory to be considered as too low'. There are buttons for 'Add', 'Update', 'Clone', 'Full clone', 'Delete', and 'Cancel' at the bottom of the table.

Figure 8: Defining a trigger function for a user macro is easy.

Compact view gives you more space for the display.

On this page you can confirm problems to indicate to other administrators that work on a solution has already begun. Instead, you could simply add a comment and – depending on the user authorization – change the *Severity* of a problem. Custom commands in the context of problems make it possible, for example, to create an entry for a specific problem with a webhook in the ticket system.

A calculated item uses the trigger expressions as a formula for computation, just like the trigger definitions. In this way, you can use multiple trigger functions both to detect problems and to generate derived metrics. These metrics can in turn be collected and used as a database in trigger expressions.

Automated Configuration

The low-level discovery [7] function is often used to monitor similar entities in the context of a host. It is based on an arbitrary item type that either outputs a specific JSON data structure itself or creates it in preprocessing (e.g., from the results of an

API query). Items, triggers, graphs and, if so desired, even hosts can then be generated automatically from these metrics with filters and rules. Low-level discovery is one of the most comprehensive features in Zabbix. It has been rounded off with increasing numbers of important enhancements in recent releases and forms the basis for dynamic monitoring with little manual intervention. The monitoring templates provided in Zabbix use low-level discovery extensively for automatic detection of filesystems, CPUs, and network interfaces. Advanced monitoring scenarios exist, as well, such as monitoring cloud resources or Kubernetes clusters.

Like all other functions in Zabbix, low-level discovery can be extended widely. It can be used for SNMP-based monitoring, WMI queries on Windows, database queries with ODBC, or custom scripts, among other things.

Alerting

Alerting in Zabbix is equally flexible and as highly customizable as data acquisition and problem detection. Actions use conditions to select

problem events on the basis of host-name and group, tags, or other criteria. They then complete operations for all events that belong to these problems.

Operations can break down problem notification into escalation levels. For example, the first notification could be emailed to the system administrator, and the second could be sent 10 minutes later as a short message to the on-call service. It is also easy to set up different temporally independent alerting paths for problems.

Zabbix refers to alerting paths as media types (*Alerts | Media types*). Email is a typical media type, for example. To support this, Zabbix offers native integration for Microsoft Teams, Slack, Rocket.Chat, Jira, GitHub, Mattermost, Opsgenie, and many more. A total of 32 notification methods are available for selection; they only need to be adapted to the use case.

If the desired service for a notification is missing, it can be defined at any time with a script or a webhook with JavaScript-based business logic. Media type definitions can be exported and imported by the web interface like many other configuration objects

Figure 9: The Problems view in Zabbix with filters enabled.

in Zabbix. This facilitates exchange in the community, but also the ability to transfer configurations between individual Zabbix instances or between a development and production environment.

Conclusions

Zabbix leaves virtually nothing to be desired for system monitoring on a functional level and supports quick and uncomplicated implementation in a wide range of environments. The open source software is backed by a stable, commercially successful company that is constantly pushing forward with the development of the product in line with market requirements. Admittedly, the learning curve with Zabbix is steep because the extensive

configuration options and interrelationships of the individual components are not always apparent at first glance. Companies that want to get the most out of the solution and benefit from the experience of experts would do well to take advantage of the training, consulting, implementation, and support services offered by experienced, certified experts. ■

Info

- [1] Zabbix container: [\[https://hub.docker.com/search?q=zabbix\]](https://hub.docker.com/search?q=zabbix)
- [2] Helm chart: [\[https://github.com/zabbix-community/helm-zabbix\]](https://github.com/zabbix-community/helm-zabbix)
- [3] JIT user provisioning: [\[https://blog.zabbix.com/just-in-time-user-provisioning-explained/25515/\]](https://blog.zabbix.com/just-in-time-user-provisioning-explained/25515/)
- [4] API documentation: [\[https://www.zabbix.com/documentation/current/en/manual/api\]](https://www.zabbix.com/documentation/current/en/manual/api)

- [5] Zabbix agent: [\[https://www.zabbix.com/documentation/current/en/manual/config/items/itemtypes/zabbix_agent\]](https://www.zabbix.com/documentation/current/en/manual/config/items/itemtypes/zabbix_agent)
- [6] Trigger functions: [\[https://www.zabbix.com/documentation/current/en/manual/appendix/functions\]](https://www.zabbix.com/documentation/current/en/manual/appendix/functions)
- [7] Low-level discovery: [\[https://www.zabbix.com/documentation/current/en/manual/discovery/low_level_discovery\]](https://www.zabbix.com/documentation/current/en/manual/discovery/low_level_discovery)

The Author

Christian Anton is Head of the Cloud Native Technologies Competence Center and a Technology Evangelist at Enthus, an association of four successful players in the German IT consulting and systems house landscape. He has more than 20 years of expertise in enterprise open source software, particularly in the areas of monitoring, automation, networking, containerization, and modern IT infrastructures.



Too Swamped to Surf?

Our ADMIN Update newsletter delivers news and technical articles to your inbox every week.

Subscribe free, and you'll discover:

- practical articles on security, cloud computing, DevOps, and more
- IT industry news
- job listings for IT professionals
- IT industry events
- and more!



bit.ly/HPC-ADMIN-Update



OpenCanary attack detection

Canary in a Coal Mine

The canary in a coal mine has made its way metaphorically into IT security with the OpenCanary honeypot for detecting attacks. By Matthias Wübbeling

The idea of using honeypots to let attackers penetrate specially prepared systems in an effort to learn more about the attackers themselves and the methods they use is long established. The aim is to uncloak the perpetrators' actions and, in particular, how they move around the network (lateral movement) or what information they access.

Honeypots are also deployed to capture email spam. Email accounts created especially for this purpose are published in non-visible areas of websites. The assumption is that bots collect these addresses and use them to spread spam. The incoming email is bound to be spam and can therefore help improve the filter.

OpenCanary [1] lets you set up network services quickly, and it notifies you when they are accessed. You have many options. On the one hand, you can run OpenCanary on public addresses that are not used for other purposes. Neighboring IP addresses of publicly available services, but also neighboring or unused ports on these machines, are often a good choice. If you are running a web server, it usually responds to requests on ports 80 and 443. Nowadays these requests are often forwarded to internal services

with the web server as a proxy. Attackers try to access poorly secured or vulnerable servers or internal information over ports 8001, 8080, 8443, or 9000, for example.

If you run a honeypot with a public IP address, you will quickly notice that an incredible number of requests are addressed there. Most are probably just scans, often triggered by (mostly) legitimate systems, such as vulnerability scanners like Shodan [2] or security researchers around the world. If you generate an alert for each event, you will miss the actual attacks in the mess of data.

OpenCanary offers genuine added value if you run it on your internal network. Once an attacker has penetrated a corporate network, they will pursue different goals, starting with persistent login options on the computer to which they already have access. Maybe they will install a reverse shell, a customized remote administration toolkit (RAT), or simply TeamViewer to access the hijacked computer time and time again.

Installation in Docker

You have different options for deploying OpenCanary on your network. If

you use pip in your Python environment, you can easily start the installation with

```
pip install opencanary
```

For individual modules, such as Samba, you will then need additional dependencies. A Docker installation on your server is easier. The OpenCanary Git repository is already prepared for use. To create your Docker container, first clone OpenCanary with the command:

```
git clone https://github.com/thinkst/opencanary.git
```

In the `opencanary/data` folder you will find the `.opencanary.conf` file, where you can configure your notifications and active services. To receive an email notification when access occurs, look for the PyLogger configuration under `logger`. You will see two preconfigured handlers: one for logging to the console and one for the `opencanary.log` logfile. **Listing 1** shows how to add another handler.

You need to adjust the port in `mail-host` to match your own mail server. In my tests, name resolution of the

mail host in the container did not always work reliably. To avoid seeing errors during your tests, just type your mail server's IP address instead of the domain.

Trial Run

If you want to add more services to those you already fired up, `ftp` and `http`, find the `.enabled` parameter in each case and change the value from `false` to `true`. In Windows environments, terminal servers (Remote Desktop Protocol, RDP) are a good choice, whereas attackers in Linux environments would try accessing SSH servers to move around the network. Once you have enabled all the services you want, close the configuration file. Before proceeding, you need to forward all the required ports in the `docker-compose.yml` file to match your choice of services; then you can create and start the container:

```
docker compose up -d --build latest
```

Now you can use `netstat` to check that Docker is using the selected ports. In the meantime, you should have a message about OpenCanary starting up in your email inbox – you configured logging for this. To only receive alerts to the defined email address, you need to set up appropriate filters with PyLogger. If you enabled the HTTP service, you can call it for a test by pointing your browser at

`https://localhost`. OpenCanary emulates a Synology disk station at this address (Figure 1).

After that, you should find mail in your inbox with a matching alert. The message only contains a JSON-formatted alert. Because it is created by PyLogger, it can also be evaluated automatically from the logfile created by OpenCanary, which is easier to integrate with your existing monitoring setup.

The logged information changes from service to service. If you use the MySQL honeypot, you will also see the user names and any passwords entered in the `logdata` field of the alert, which could already give you a first hint as to the path and possibly the account the attacker is using. The most important information in an alert, however, is likely to be the `src_host`. Although that is simply the Docker host in this example, in a real attack on your systems, you will see an IP address in your infrastructure. The system with this IP address is probably already controlled by some attacker.

The response to detecting an attacker can even be automated to some extent if you use dynamic network configurations. You can isolate the affected computer from the network and let it continue to run in a protected environment – a walled garden – for the time being and examine it more closely without the attacker directly noticing that they have already

Listing 1: Adding a Handler

```
"SMTP": {
  "class": "logging.handlers.SMTPHandler",
  "mailhost": ["linux-magazine.com", 587],
  "fromaddr": "canary@linux-magazine.com",
  "toaddrs": ["alert@linux-magazine.com"],
  "subject": "Alert from OpenCanary",
  "credentials": ["canary@it-administrator",
  "password"],
  "secure": []
}
```

been discovered. Otherwise, you will want to remove the affected system completely from the network for further investigation.

Conclusions

OpenCanary helps you track down attackers as they roam your network. The services available out of the box cover the type of servers that usually exist in a corporate environment and are unlikely to attract excessive scrutiny from attackers. As shown in this article, it is easy to deploy an initial version of the honeypot framework on your network and receive alerts by email. From there, you can continue to strengthen your company's line of defense. ■

Info

- [1] OpenCanary: <https://opencanary.readthedocs.io/en/latest/>
- [2] Shodan: <https://www.shodan.io>

The Author

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

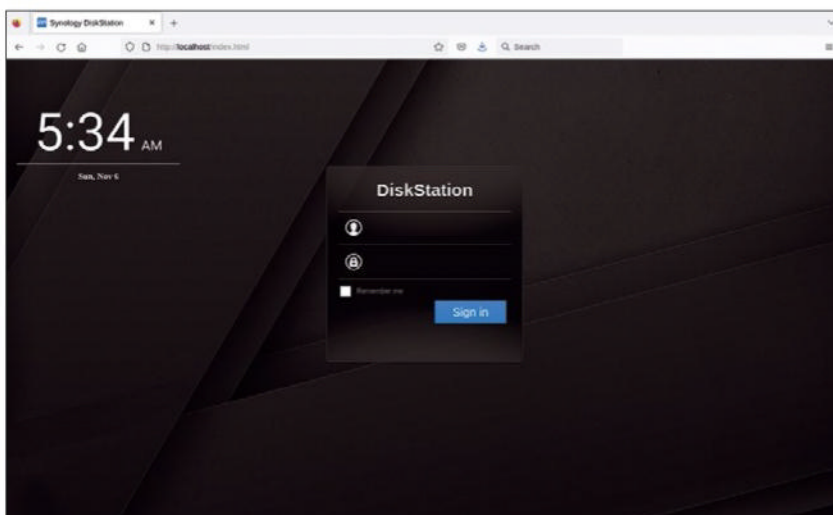
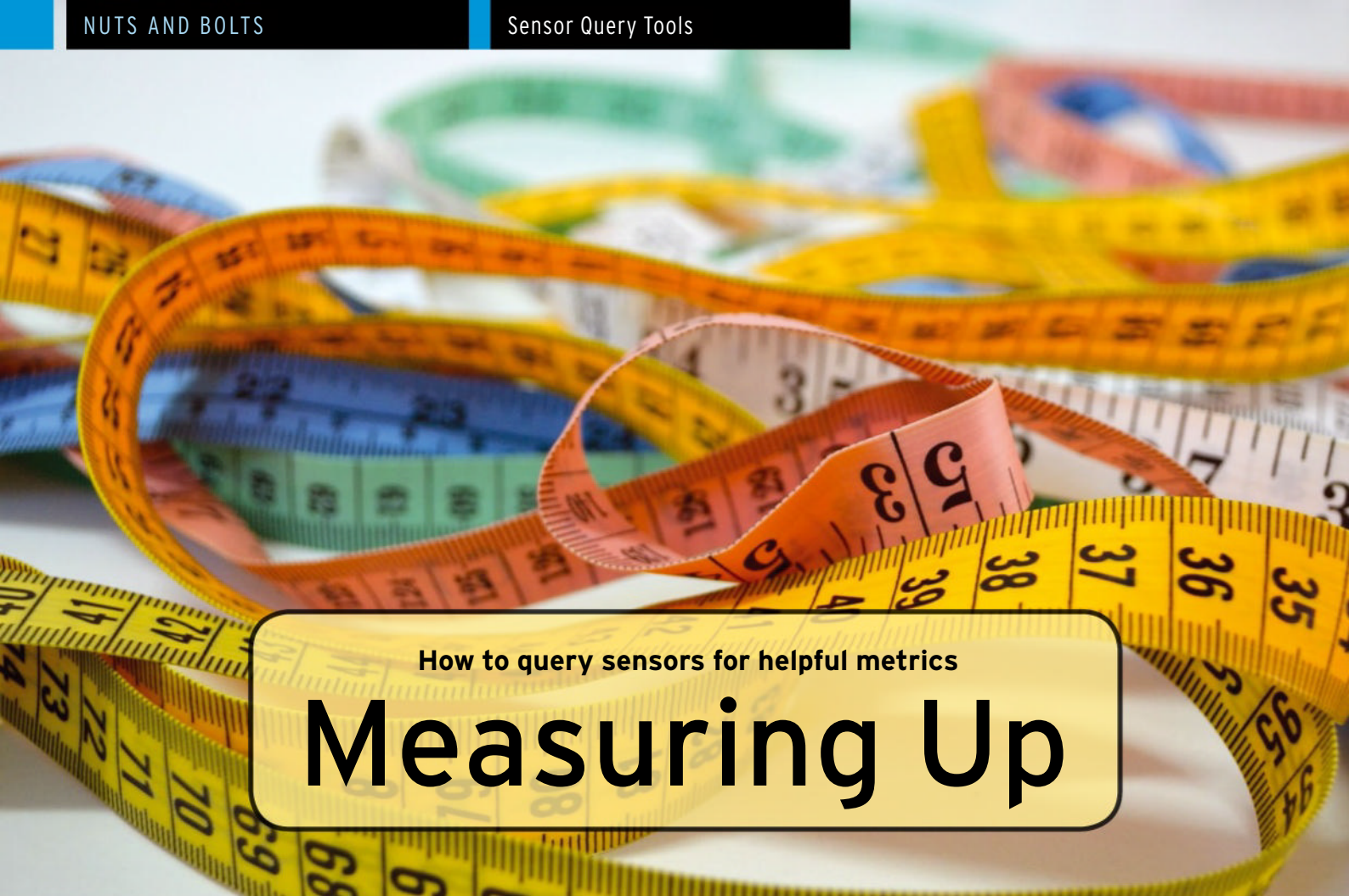


Figure 1: A Synology DiskStation login generated by OpenCanary.



How to query sensors for helpful metrics

Measuring Up

Discover the sensors that already exist on your systems, learn how to query their information, and add them to your metrics dashboard.

By Andreas Stolzenberger

Metrics dashboards in Grafana or similar tools show you how your infrastructure is performing. Querying sensors gives you additional information (e.g., voltages, temperatures, and fan speeds) that you can use to analyze and protect against failures. In this article, I look at the tools you can use to query sensors.

Previous articles looked into the use of tools such as InfluxDB, Telegraf, or Collectd to help retrieve metrics from running systems and visualize the results in Grafana. Thus far, however, the articles have focused on the basic software architecture of a TICK stack [1] (i.e., Telegraf, InfluxDB, Chronograf, Kapacitor; alternatively, TIG [2] with Grafana instead of Chronograf, or CIG with Collectd instead of Telegraf) and how to use it to collect performance data from your environment. In other words, you already know the utilization level of your mass storage media and the kind of performance the CPUs deliver. Besides plain vanilla performance

data such as system load, it is a good idea to keep track of other data such as temperatures and voltages, because these have a significant influence on the durability of the server components.

Of course, you could install external sensors and temperature probes in your data center and query them, but before you invest in external devices, it's worth taking a look at the sensors that already exist in the system and discovering how to query their information and add them to your metrics dashboard. This approach is also recommended for users who use rented servers from a hardware provider and do not have physical access to their data centers.

Sensor monitoring can turn up some surprises: For example, when the disk temperatures in a RAID array reveal that one of four disks is permanently 20 degrees warmer than all the others, it explains why the drive failed twice within a year and had to be replaced. Because the user had the

recorded metrics, they were given a replacement machine.

If you do not run a TIG stack in your data center or on your cloud server, you can employ a simple solution that relies on containers with Podman.

TIG with Podman

An environment with Grafana and InfluxDB can be set up very quickly with containers. Systemd handles the management tasks; the containers act like system services and are automatically started by `systemctl` and enabled directly after the system boots. In this scenario, the components operate independently and on their own IP addresses. Therefore, you can try out different constellations and use several versions of InfluxDB at the same time.

For the containers to work with their own IP addresses, you need a server with a network bridge and a Podman network of the Bridge [3] type. You will find various how-tos online to help you create a bridged container network of the `macvlan` type. However, if you use this type of network, the containers cannot communicate with the host itself.

Photo by patricia serna on Unsplash

In this test setup, the bridge LAN is named `pub_net` and is described by the `/etc/cni/net.d/pub_net.conf` list file. The setup runs on a server with RHEL 8, CentOS 8 Stream, or an Enterprise Linux 8 (EL8) clone and with the Podman package installed.

InfluxDB with Flux

Version 2 of InfluxDB introduced a number of significant changes, including the Flux query language as the default. Many users still prefer InfluxQL version 1.x, which also makes integration with Grafana far easier. Unfortunately, Grafana still does not offer a convenient graphical query editor for Flux. By the way, a simple trick will help Flux newcomers. The new InfluxDB web user interface (UI) provides a graphical query editor itself. Build your query there and then copy the resulting Flux code into the Grafana panel.

The setup with Podman allows for worry-free use of both variants in this workshop, which means you can take your time and get to know both versions before deciding which one you want to use in production. For Influx version 1.8, create a systemd service in the `/etc/systemd/system/flux18.service` file following the template in [Listing 1](#).

Now when you type

```
systemctl start flux18
```

systemd creates two directories under `/var/pods/flux18` where the container will store the configuration (`etc`) and the data (`data`), which means the information is retained on the host system even after the container is stopped or restarted. You can define the container's IP address and the MAC address statically. For tests with InfluxDB 2.0, create an `/etc/systemd/system/flux24.service` file with exactly the same template. Change the IP and MAC address (here, to `.82` and `:52`), swap `flux18` in the container name and directories for `flux24`, then change the two lines in the Docker template from `docker.io/influxdb:1.8` to `docker.io/influxdb:2.4`. Because

InfluxDB 2.0 by default no longer allows access without a login, you need to click on `http://192.168.2.82:8086` in the browser after starting the container and run through the basic setup wizard. Create two API tokens for Telegraf and Grafana at the same time.

Following the pattern, create a service declaration for your Grafana container. The name is `grafana`; the image points to `docker.io/grafana/grafana`. The two directories for the Grafana container are `logs` and `data`:

```
--volume /var/pods/grafana/data:Z
/var/lib/grafana:Z Z
--volume /var/pods/grafana/logs:Z
/var/log/grafana:Z
```

Moreover, you can specify at the start of the file that you want systemd to start the Grafana container after the Influx container at system boot:

```
After=network-online.target flux18.service
```

In the test setup, Grafana has an IP address of `.80` and a MAC address of `:50`. With Grafana, too, you first need to go through the initial setup in the browser at `http://192.168.2.80:3000`.

Sensors in the System

Depending on your hardware, most motherboards and chipsets already have a whole range of built-in sensors that can be queried by the operating system. The toolset for this is the Linux management sensors (*lm-sensors* suite) and can be set up on Enterprise Linux systems by typing

```
yum install lm_sensors
```

(or using `dnf`). To help you with existing sensors, the package includes a `sensors-detect` tool that detects the chipset sensors, the I2C buses, and the sensors attached to them; it writes the initial module configuration to the `/etc/sysconfig/lm_sensors` file. A call to the `sensors` tool lists the sensors and their values. Depending on the system, some of them might not be usable. In some places, *lm-sensors*

Listing 1: Influx v1.8 systemd Service

```
[Unit]
Description=Influxdb 1.8
After=network-online.target
Wants=network-online.target

[Service]
ExecStartPre=mkdir -p /var/pods/flux18/etc
ExecStartPre=mkdir -p /var/pods/flux18/data
ExecStartPre=/bin/podman kill flux18
ExecStartPre=/bin/podman rm flux18
ExecStartPre=/bin/podman pull docker.io/influxdb:1.8
ExecStart=/bin/podman run
--name flux18
--volume /var/pods/flux18/etc:/etc/influxdb:Z
--volume /var/pods/flux18/data:/var/lib/influxdb:Z
--net pub_net
--ip 192.168.2.81
--mac-address 52:54:00:A8:02:51 \docker.io/influxdb:1.8
ExecStop=/bin/podman stop flux18

[Install]
WantedBy=multi-user.target
```

finds sensors that do not even exist in the system, often because the hardware manufacturer has installed a connector and a controller, but not the sensor itself. Of course, these fake sensors are immediately apparent, because case temperatures of 0 or -273 degrees are pretty unlikely.

In practice, however, sensors will at least find the core package on almost all systems, which means the temperature information for the CPU chip and all CPU cores it contains. Already this gives you some important insights and is especially important for passively cooled edge devices or servers with a small form factor (e.g., Intel NUC). If the system does not have any fan speed sensors, the CPU temperature at least lets you draw conclusions about fan problems ([Figure 1](#)).

Monitoring Hard Disks

You need to pay special attention to monitoring your hard disks. It does not matter whether they are old-fashioned mechanical drives or modern SSDs; you have a number of important metrics to capture. Of course, temperature plays a big role in mechanical drives. If the server does not have a separate temperature sensor, you can use the disk temperature

to draw conclusions about the room temperature in the data center itself. If, for example, the air conditioning in a server facility fails, this quite quickly becomes apparent through the rise in disk temperature. Both legacy disks and SSDs have a SMART function that attempts to predict mass storage defects and failures. Information on read and seek errors is important, indicating problems with individual drives or the complete array. Two main tools can discover information about drive temperatures on Linux servers: `hddtemp` and `smartctl`. As the name suggests, `hddtemp` returns only the temperature of a drive, whereas `smartctl` also returns data on error rates and SMART status. However, SMART values should be taken with a grain of salt, because different hard drive manufacturers give you very different values. In my lab setup, for example, I get consistently high values for seek and read errors from Seagate Enterprise disks, whereas Toshiba drives only return 0. Seagate outputs the raw error rates before internal error correction. The SMART specification does not specify exactly what the output values return. Because both tools require

root privileges to query the information, metrics collectors such as Telegraf, which runs in userspace, can only access the data in a roundabout way. I'll talk about this later when it comes to sending the collected values to Influx. Type

```
dnf install hddtemp
dnf install smartmontools
```

to install the tools on an EL8 system.

Intelligent Platform Management with IPMI

Server manufacturers typically build a baseboard management controller (BMC) chip into their servers, which then provides an Intelligent Platform Management Interface (IPMI). This interface gives you very detailed information about the system, which the BMC collects independent of the operating system. Depending on the implementation, the IPMI can be addressed on the local system or on the LAN. What can be retrieved by IPMI depends on the BMC used and differs depending on the vendor and age of the server. I used different systems for this article: an Intel NUC and a cloud

server by Hetzner, which does not have a BMC.

An older HP Gen8 microserver at least provides a few temperature values for the RAM, chipset, and case, whereas its big brother, the DL380 Gen8, reveals the current power consumption of the power supply unit and reams of temperature data from every imaginable place in the chassis. Newer Dell and Fujitsu servers also report very detailed fan speeds as well as chipset, DIMM, and processor voltages and currents. You need `ipmitool` to access the local IPMI, which with the commands

```
dnf install ipmitool
ipmitool sdr elist
```

sets up and lists available sensors and values.

Collecting Data for the UPS

An uninterruptible power supply (UPS) is essential in any server facility. Depending on the design and monitoring port, you can also glean valuable information about the condition of this device. It is not just about battery life and utilization. The

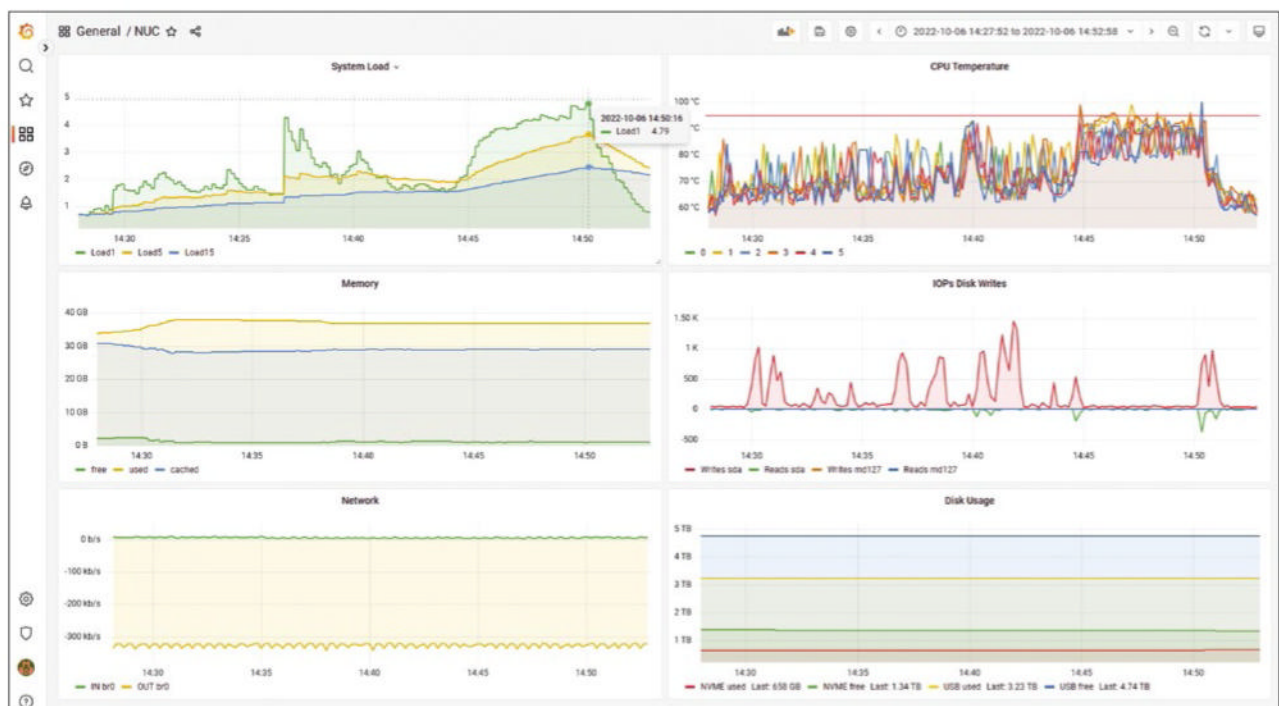


Figure 1: The sensor evaluation of the processor temperature reveals that the cores overheat because the Core i7 cooling in an Intel NUC is not sufficient at higher system loads.

voltage curve also provides important information. Machine crashes or the unexpected death of a component (e.g., a hard disk) are often accompanied by voltage fluctuations in the mains supply.

Large UPSs with a LAN management adapter disclose this basic and other information by way of the Simple Network Management Protocol (SNMP) – but more of that in a moment. A simple UPS comes without a LAN interface, but with a serial or USB interface for monitoring. Often the Network UPS Tools (NUT) suite is then used to monitor the UPS and organize controlled shutdowns of several systems on the LAN. The `upsd` UPS daemon from the NUT toolset permanently contacts the UPS system and continuously queries the metrics. If you have the right metrics grabbers, this information can also be routed to Influx and the Grafana dashboard.

Querying by SNMP

Many active components on the LAN support SNMP, which is rarely used today to manage the components actively, but it is excellent for querying and collecting metrics (Figure 2). SNMP protocol versions 1 and 2 are considered insecure. However, almost all devices with SNMP restrict the operation of these protocols to the LAN

and only support read-only mode.

With the `net-snmp` package installed on the system, Telegraf can read individual SNMP values or complete tables and forward them to InfluxDB. Which values these are usually depends on the manufacturer's own SNMP management information base (MIB). The MIB provides information on the existing object IDs (OIDs) and their addresses. Detailed documentation of custom SNMP MIBs can be found on the vendor websites.

In the example with a UPS by APC (Schneider Electric), a suitable configuration for the manufacturer's custom MIB is shown in Listing 2. The first `inputs.snmp` entry opens the UDP connection to the management adapter. The `inputs.snmp` entries that follow gather information on the battery temperature and the input and output voltages. You can also field similar metrics from switches or routers this way.

Collecting Data with Telegraf

You have many different ways to retrieve sensor data from the tools and send the data to Influx. For example, you can write your own scripts that either communicate directly with the InfluxDB API on port 8068 of the InfluxDB server or send queries by the InfluxDB client.

Listing 2: UPS Metrics

```
[[inputs.snmp]]
  agents = ["<IP address of the UPS management boards>:161"]
  version = 2
  community = "public"
  name = "snmp"
[[inputs.snmp.field]]
  name = "Battery Temperature"
  oid = "1.3.6.1.4.1.318.1.1.2.2.2.0"
[[inputs.snmp.field]]
  name = "PowerIN"
  oid = "1.3.6.1.4.1.318.1.1.3.2.1.0"
[[inputs.snmp.field]]
  name = "PowerOUT"
  oid = "1.3.6.1.4.1.318.1.1.4.2.1.0"
```

In most cases, however, metric collectors such as Telegraf or Collectd are used on the systems. Both come with input modules that directly support all of the sensor sources presented above. In the Telegraf configuration, before the input modules, the `/etc/telegraf/telegraf.conf` file is preceded by the output. In line with the example, you want the host to be monitored to deliver its data to both the InfluxDB 1.8 and 2.4 hosts. For Influx 1.8, that entry would be:

```
[[outputs.influxdb]]
  url = "http://192.168.2.81:8086"
  database = "telegraf"
```

Things are a little more complicated with InfluxDB 2.4: You must first

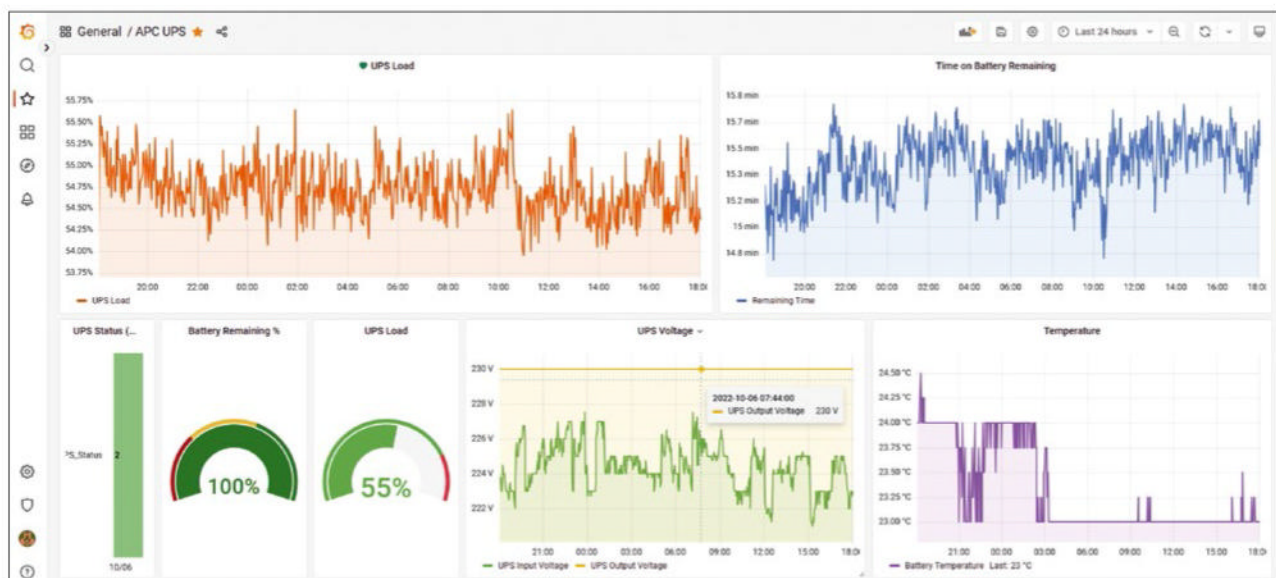


Figure 2: Sensor data is being fed by SNMP into the metrics dashboard from the online UPS. Unlike a standby UPS, the double converter filters out all mains fluctuations.

create an API token, the organization, and a bucket in the basic configuration of the InfluxDB pod:

```
[[outputs.influxdb_v2]]
  urls = ["http://192.168.2.82:8086"]
  token = "<token>"
  organization = "<organization>"
  bucket = "<bucket>"
```

If you specify both outputs in your `telegraf.conf` file, Telegraf will also send all metrics to both Influx pods. Alternatively, you could use Prometheus instead of InfluxDB. To avoid going beyond the scope of the workshop, I will not look at this option at this point. You can also append the input sources directly in `/etc/telegraf/telegraf.conf`.

For debugging purposes, however, it is a good idea to create a separate CONF file for each input in `/etc/telegraf/telegraf.d/`, then you can very easily type

```
telegraf --test 2
--config /etc/telegraf/2
telegraf.d/<sensor>.conf
```

to test the output of the sensor sources individually and check the supplied data.

To use `lm-sensors`, you simply need a CONF file named `sensors.conf` with a one-liner: `[[inputs.sensors]]`. Of course, you do need to have installed and configured the `lm_sensors` toolset on the system up front. The same is true for IPMI input. The `[[inputs.ipmi_sensor]]` line in the appropriate CONF file is all you need.

Things get a little more complex with SMART or `hddtemp`. Both tools need root privileges, but the Telegraf client works in userspace and therefore cannot access the devices in `/dev`. The `hddtemp` tool works around the problem in an elegant way with its own daemon, which you can enable by typing

```
systemctl enable hddtemp --now
```

The daemon then serves up the queried disk temperatures in userspace over a simple TCP port on

`localhost:7634`. With the `hddtemp` service running, the one-liner `[[inputs.hddtemp]]` in the `hddtemp.conf` file is all you need.

SMART, on the other hand, does not work around root privileges. To use this input, you need to give the `telegraf` account root privileges for the `smartctl` tool by creating a `/etc/sudoers.d/telegraf` file with the following content:

```
telegraf ALL = 2
NOPASSWD: /usr/sbin/smartctl
```

You can then enable the input in the Telegraf `smart.conf` configuration file:

```
[[inputs.smart]] use_sudo = true
```

Unfortunately, `smartctl` caused problems in my various lab setups, especially on the HP microserver with a low-powered Celeron CPU. The SMART input caused extremely high CPU loads, affecting the running services. Only `hddtemp` is used on this system.

To retrieve the values of a UPS managed by NUT, you need an IP connection to the NUT server and a suitable monitoring user, including a password, which you have defined on the NUT server in `/etc/ups/upsd.user`. The entry in the `/etc/telegraf/telegraf.d/ups.conf` file is then:

```
[[inputs.upsd]]
  server = <server IP address>
  port = 3493
  username = "<ups-user>"
  password = "<ups-user-password>"
```

If NUT and Telegraf are running on the same host, leave out the server line.

Telegraf Without a Plugin

The choice of plugins for Telegraf is huge, but you could still be confronted with proprietary metrics sources of which Telegraf is not aware. You could, on the one hand, use a script that communicates directly with the InfluxDB API, as mentioned at the beginning.

However, if you want to send data by Telegraf only, you can use its `[[input.exec]]` option instead. Telegraf then runs a given script (Bash, Python, Perl, etc.) and routes the data back to InfluxDB. The `input.exec` file expects the response data in the `<name> = <value>` format; for example:

```
room temperature kitchen=20
```

A matching entry in the Telegraf configuration would then be:

```
[[inputs.exec]]
  commands = [2
    "sh /etc/telegraf/script/script1.sh"
  ]
  timeout = "5s"
  data_format = "influx"
```

It is important for Telegraf to execute the custom script in its userspace (i.e., with rights of the `telegraf` user). Here, users sometimes get into a spin when testing their inputs. When running

```
telegraf --test
```

with root privileges, everything works, whereas the Telegraf daemon running in the `telegraf` user context suddenly stops delivering values. Make sure you run your tests in the right context.

Collectd Directly or with Detours

As a metric collector, Collectd takes a virtually identical approach to Telegraf, at least if you use InfluxDB version 1.8. This version accepts metrics from Collectd directly. In this example, you will find the InfluxDB configuration for the InfluxDB 1.8 container in the `/var/pods/flux18/etc/influxdb.conf` directory on the host:

```
[[collectd]]
  enabled = true
  bind-address = "192.168.2.81:25826"
  database = "collectd"
  typesdb = 2
    "/usr/share/collectd/types.db"
```

As of version 2, InfluxDB no longer has direct Collectd input. If you collect metrics with this tool, you have to detour by way of a Telegraf instance that is configured as,

```
[[inputs.socket_listener]]
  service_address = "udp://:25826"
  data_format = "collectd"
```

which is followed by `[[outputs.influxdb_v2]]`, as described earlier in the text. Collectd provides all of the functions mentioned so far for inputs. A config example in `/etc/collectd.conf` is

```
LoadPlugin hddtemp
<Plugin hddtemp>
  Host "127.0.0.1"
  Port "7634"
</Plugin>
<Plugin network>
```

```
Server "192.168.2.81" "25826"
</Plugin>
```

in which Collectd queries `hddtemp` and returns the values to the Influx 1.8 server.

Conclusions

The plain old performance metrics of your server and cloud systems appear in a different context when you have access to sensor data. Suddenly, you can link performance drops of a server to an overheated CPU and perhaps even find the root of this problem in the form of a fan that is too slow. You will then also see disk failures in the context of power fluctuations or surges.

Besides plain vanilla monitoring and visualization of sensor data, you can

also link the data to alerting and system management. If the intake temperature of any of your servers rises to above 40 degrees, you probably have a problem with the air conditioning in the server facility, and you will want to shut down all the systems – or at least the less important ones – to protect your hardware investment. ■

Info

- [1] Monitoring, alerting, and trending with the TICK Stack: <https://www.admin-magazine.com/Archive/2018/47/Monitoring-alerting-and-trending-with-the-TICK-Stack/>
- [2] Storage monitoring with Grafana: <https://www.admin-magazine.com/Archive/2019/54/Storage-monitoring-with-Grafana/>
- [3] Podman network: <https://docs.podman.io/en/latest/markdown/podman-network-create.1.html>

IT Highlights at a Glance



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

Linux Update • ADMIN Update • ADMIN HPC

Keep your finger on the pulse of the IT industry.

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

Linux Update: bit.ly/Linux-Update



Creating load for fun and profit

Stressing Out

Standard loads are essential to benchmarking.

By Federico Lucifredi

I have discussed load-generating tools on these pages from the very beginning [1] out of simple practical necessity. Setting aside the benchmarks themselves, the facility generating a load consistently over and over again is the most important tool in a performance tester's arsenal. Amos Waterland's stress [2] has long been my go-to tool because of its simple, reliable effectiveness, but I have been watching with interest Colin King's promising stress-ng [3] since our time together on the Ubuntu Server team. Time to take the challenger out for a spin!

The New King

The newer stress-ng tool can be promptly installed with

```
sudo apt install stress-ng
```

in Ubuntu Server versions dating as far back as 2015. Insuring a system is exercised consistently during testing is the tool's primary mission, validating software response as well as hardware reliability and its behavior under load. The latest release provides a menagerie of 300 stressors crafted to

load the CPU (with a variety of more than 80 tests), memory and CPU caches, filesystems, and even the virtual memory system.

The original stress syntax remains valid, and one can start up resource stressors in the same fashion. Figure 1 shows one such example, including the first notable difference. The tool itself originated as a load generator, but it also outputs results that can be construed as a benchmark of sorts. The maintainer advises against using stress-ng as a precise benchmark on the project's web page, but he also points out that these metrics can be useful to monitor relative regression between operating system (OS) versions, particularly when used on the same hardware.

Load figures are provided in arbitrary "Bogo Ops" [4] units (bogus operations per second) and will run for a default period of 24 hours unless instructed otherwise (with the -t or --timeout options). Bogo Ops are defined in the context of a specific stressor and should not be compared across different resource loads, but they absolutely can be compared within the same stressor. While not aspiring to be perfect benchmarks,

they do make a convenient starting point.

Complex Loads

I am not running the new tool to use the old loads. A good example of what the 300 stressors include is matrix, notorious for being the stressor expected to heat Intel processors the most by stressing the CPU with floating-point arithmetic while exercising its caches and the system's memory:

```
stress-ng --matrix 0 -t 1m --tz --times
```

The zero count syntax requests one stressor to run on each CPU in the system, --times generates statistics on userland and kernel time (Figure 2), and the --tz option collects CPU temperature data where available. It is also possible to collect detailed statistics with perf [5] by using the --perf option:

```
stress-ng --matrix 0 -t 1m --perf
```

Figure 3 shows the resulting output. This type of analysis is more interesting when tuning code as a developer

rather than as an operator, and attempting to improve cache performance and keeping page faults in check is of interest. The stress-ng tool is remarkably good

```
$ stress-ng --cpu 4 --vm 2 --hdd 1 --timeout 30s --metrics
stress-ng: info: [2394] setting to a 30 second run per stressor
stress-ng: info: [2394] dispatching hogs: 4 cpu, 2 vm, 1 hdd
stress-ng: info: [2394] successful run completed in 30.28s
stress-ng: info: [2394] stressor      bogo ops real time  usr time  sys time  bogo ops/s  bogo ops/s  CPU used per
stress-ng: info: [2394]                (secs)    (secs)    (secs)    (real time) (usr+sys time) instance (%)
stress-ng: info: [2394] cpu          11380    30.20    16.67    0.03    376.80    681.44    13.82
stress-ng: info: [2394] vm          298472    30.10    5.82    2.48    9915.65    35960.48    13.79
stress-ng: info: [2394] hdd         43945    30.27    2.45    1.76    1451.60    10438.24    13.91
$
```

Figure 1: The new tool can include load statistics in its output.

at tailored cache stress plans – better than anything else I have seen to date. Another interesting aspect is the possibility to load a stressor to a fixed share of CPU core, as opposed to maxing out a core with 100 percent load. For example,

```
stress-ng -t 1m -c 0 -l 40
```

will burden all cores in the system (-c 0) with a 40 percent load (Figure 4). Loading all cores to 40 percent is something not trivially accomplished without tooling. To torment the hardware, stress-ng can vary its approach within two abstractions: methods and classes. Methods

are located within a given stressor and are executed round-robin as time permits. The command

```
stress-ng --cpu-method which
```

lists more than 50 methods to load a CPU. When consistency is desirable, you can select a given method individually (--cpu-method).

Classes are the second such facet.

Stressors are grouped together in different classes according to the subsystems they load the most – CPU, memory, I/O, network, and so on. All tests in the class can be run in parallel or, perhaps more usefully, sequentially, as in this example,

```
stress-ng --class network --seq 0
```

which runs all network tests sequentially, one instance per CPU. One last consideration is that stress-ng is great at creating memory pressure. Try running the following command with swap turned off (swapon -a) and watch the fireworks:

```
stress-ng --brk 2 --stack 2 --bigheap 2
```

Colin has presented his work at several Linux Foundation events, and slides [6] and recordings [7] are readily available to those who want to learn more about this great tool. ■

```
federico@ferenginar:~$ stress-ng --matrix 0 -t 1m --tz --times
stress-ng: info: [19298] dispatching hogs: 2 matrix
stress-ng: info: [19298] cache allocate: default cache size: 1024K
stress-ng: info: [19298] successful run completed in 60.00s (1 min, 0.00 secs)
stress-ng: info: [19298] thermal zone temperatures not available
stress-ng: info: [19298] for a 60.00s run time:
stress-ng: info: [19298] 120.00s available CPU time
stress-ng: info: [19298] 119.72s user time ( 99.76%)
stress-ng: info: [19298] 0.00s system time ( 0.00%)
stress-ng: info: [19298] 119.72s total time ( 99.76%)
stress-ng: info: [19298] load average: 1.39 0.52 0.22
federico@ferenginar:~$
```

Figure 2: Measuring the results of the best load to raise CPU temperature.

```
federico@ferenginar:~$ stress-ng --matrix 0 -t 1m --perf
stress-ng: info: [19306] dispatching hogs: 2 matrix
stress-ng: info: [19306] cache allocate: default cache size: 1024K
stress-ng: info: [19306] successful run completed in 60.00s (1 min, 0.00 secs)
stress-ng: info: [19306] matrix:
stress-ng: info: [19306] 263,173,613,704 CPU Cycles 4.39 D/sec
stress-ng: info: [19306] 430,950,264,576 Instructions 7.18 D/sec (1.638 instr. per cycle)
stress-ng: info: [19306] 109,731,350,942 Cache References 1.83 D/sec
stress-ng: info: [19306] 5,423,268 Cache Misses 90.39 K/sec ( 0.00%)
stress-ng: info: [19306] 38,661,029,628 Stalled Cycles Frontend 0.64 D/sec
stress-ng: info: [19306] 41,506,068,272 Stalled Cycles Backend 0.60 D/sec
stress-ng: info: [19306] 54,523,992,442 Branch Instructions 0.91 B/sec
stress-ng: info: [19306] 440,401,482 Branch Misses 7.34 M/sec ( 0.81%)
stress-ng: info: [19306] 98 Page Faults Minor 1.63 sec
stress-ng: info: [19306] 0 Page Faults Major 0.00 sec
stress-ng: info: [19306] 30,926 Context Switches 182.10 sec
stress-ng: info: [19306] 0 CPU Migrations 0.00 sec
stress-ng: info: [19306] 0 Alignment Faults 0.00 sec
federico@ferenginar:~$
```

Figure 3: Branch misses, page faults, cache misses, and even alignment faults are detailed.

```
top - 14:49:42 up 23 days, 23:38, 2 users, load average: 0.69, 0.29, 0.20
Tasks: 143 total, 1 running, 142 sleeping, 0 stopped, 0 zombie
%Cpu(s): 42.2 us, 0.2 sy, 0.0 ni, 57.6 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 8045804 total, 6911924 free, 125028 used, 1008852 buff/cache
KiB Swap: 8247292 total, 8247292 free, 0 used, 7531836 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
19323	federico	20	0	32608	5328	3308	S	43.5	0.1	0:17.74	stress-ng-cpu
19322	federico	20	0	32608	5332	3316	S	40.2	0.1	0:17.79	stress-ng-cpu
19324	federico	20	0	41796	3628	3056	R	0.3	0.0	0:00.11	top
1	root	20	0	119972	6028	3908	S	0.0	0.1	0:18.83	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.17	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:39.53	ksoftingd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
7	root	20	0	0	0	0	S	0.0	0.0	5:34.21	rcu_sched

Figure 4: stress-ng brackets the target load when looking through top.

Info

- [1] "Performance Tuning Dojo" by Federico Lucifredi, *ADMIN*, issue 8, 2012, pp. 90-92
- [2] Amos Waterland - stress: [\[https://manpages.ubuntu.com/manpages/jammy/man1/stress.1.html\]](https://manpages.ubuntu.com/manpages/jammy/man1/stress.1.html)
- [3] Colin King - stress-ng: [\[https://github.com/ColinIanKing/stress-ng\]](https://github.com/ColinIanKing/stress-ng)
- [4] Bogo Ops in stress-ng: [\[https://wiki.ubuntu.com/Kernel/Reference/stress-ng#Bogo_Ops\]](https://wiki.ubuntu.com/Kernel/Reference/stress-ng#Bogo_Ops)
- [5] perf: [\[https://manpages.ubuntu.com/manpages/jammy/man1/perf.1.html\]](https://manpages.ubuntu.com/manpages/jammy/man1/perf.1.html)
- [6] Colin King slides: [\[https://project.linuxfoundation.org/hubfs/Webinars/Webinar_Slides/Colin-Ian-King-Mentorship-Stress-ng.pdf\]](https://project.linuxfoundation.org/hubfs/Webinars/Webinar_Slides/Colin-Ian-King-Mentorship-Stress-ng.pdf)
- [7] Colin King webinar: [\[https://www.linuxfoundation.org/webinars/stress-ng-how-to-stress-test-your-computer\]](https://www.linuxfoundation.org/webinars/stress-ng-how-to-stress-test-your-computer)

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*.



FOSSLIFE

Open for All

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

FOSSlife.org



ADMIN

Network & Security

NEWSSTAND

Order online:
bit.ly/ADMIN-Newsstand

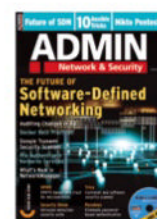
ADMIN is your source for technical solutions to real-world problems. Every issue is packed with practical articles on the topics you need, such as: security, cloud computing, DevOps, HPC, storage, and more! Explore our full catalog of back issues for specific topics or to complete your collection.

#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

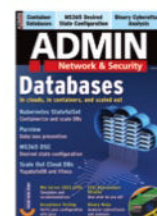


#73 - January/February 2023

Databases

Cloud databases can be useful in virtually any conceivable deployment scenario, come in SQL and NoSQL flavors, and harmonize well with virtualized and containerized environments.

On the DVD: Manjaro 22.0 Gnome



#72 - November/December 2022

OpenStack

Find out whether the much evolved OpenStack is right for your private cloud.

On the DVD: Fedora 36 Server Edition



#71 - September/October 2022

Kubernetes

We show you how to get started with Kubernetes, and users share their insights into the container manager.

On the DVD: SystemRescue 9.04



#70 - July/August 2022

Defense by Design

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

On the DVD: Rocky Linux 9 (x86_64)



#69 - May/June 2022

Terraform

After nearly 10 years of work on Terraform, the HashiCorp team delivers the 1.0 version of the cloud automation tool.

On the DVD: Ubuntu 22.04 "Jammy Jellyfish" LTS server Edition



WRITE FOR US

Admin: Network and Security is looking for good, practical articles on system administration topics. We love to hear from IT professionals who have discovered innovative tools or techniques for solving real-world problems.

Tell us about your favorite:

- interoperability solutions
- practical tools for cloud environments
- security problems and how you solved them
- ingenious custom scripts

- unheralded open source utilities
- Windows networking techniques that aren't explained (or aren't explained well) in the standard documentation.

We need concrete, fully developed solutions: installation steps, configuration files, examples – we are looking for a complete discussion, not just a “hot tip” that leaves the details to the reader.

If you have an idea for an article, send a 1-2 paragraph proposal describing your topic to: edit@admin-magazine.com.



Authors

Amber Ankerholz	6
Christian Anton	76
Frank Blessing	50
Stefan Christoph	50
Ken Hess	3
Thomas Joos	10, 64
Christian Knerrmann	44, 54
Petros Koutoupis	30
Martin Gerhard Loschwitz	72
Federico Lucifredi	94
Tufan Özdoğan	50
Julien Pivotto	68
Dr. Holger Reibold	14, 58
Christian Schulenburg	18
Artur Skura	37
Andreas Stolzenberger	20, 88
Matthias Wübbeling	26, 86

Contact Info

Editor in Chief

Joe Casad, jcasad@linuxnewmedia.com

Managing Editors

Rita L Sooby, rsooby@linuxnewmedia.com
Lori White, lwhite@linuxnewmedia.com

Senior Editor

Ken Hess

Localization & Translation

Ian Travis

News Editor

Amber Ankerholz

Copy Editors

Amy Pettie, Aubrey Vaughn

Layout

Dena Friesen, Lori White

Cover Design

Dena Friesen, Illustration based on graphics by omnimages, 123RF.com

Advertising

Brian Osborn, bosborn@linuxnewmedia.com
phone +49 8093 7679420

Publisher

Brian Osborn

Marketing Communications

Gwen Clark, gclark@linuxnewmedia.com
Linux New Media USA, LLC
4840 Bob Billings Parkway, Ste 104
Lawrence, KS 66049 USA

Customer Service / Subscription

For USA and Canada:
Email: cs@linuxnewmedia.com
Phone: 1-866-247-2802
(Toll Free from the US and Canada)

For all other countries:
Email: subs@linuxnewmedia.com
www.admin-magazine.com

While every care has been taken in the content of the magazine, the publishers cannot be held responsible for the accuracy of the information contained within it or any consequences arising from the use of it. The use of the DVD provided with the magazine or any material provided on it is at your own risk.

Copyright and Trademarks © 2023 Linux New Media USA, LLC.

No material may be reproduced in any form whatsoever in whole or in part without the written permission of the publishers. It is assumed that all correspondence sent, for example, letters, email, faxes, photographs, articles, drawings, are supplied for publication or license to third parties on a non-exclusive worldwide basis by Linux New Media unless otherwise stated in writing.

All brand or product names are trademarks of their respective owners. Contact us if we haven't credited your copyright; we will always correct any oversight.

Printed in Nuremberg, Germany by Zeitfracht GmbH.

Distributed by Seymour Distribution Ltd, United Kingdom

ADMIN is published bimonthly by Linux New Media USA, LLC, 4840 Bob Billings Parkway, Ste 104, Lawrence, KS 66049, USA (Print ISSN: 2045-0702, Online ISSN: 2831-9583). May/June 2023. Periodicals Postage paid at Lawrence, KS. Ride-Along Enclosed. POSTMASTER: Please send address changes to ADMIN, 4840 Bob Billings Parkway, Ste 104, Lawrence, KS 66049, USA.

Represented in Europe and other territories by: Sparkhaus Media GmbH, Bialasstr. 1a, 85625 Glonn, Germany.

Looking for your place in open source?



Set up job alerts and get started today!

OpenSource JOB HUB



opensourcejobhub.com/jobs



i 16:10-16 inch display und maximum sized 99 Wh battery. Compatible with the TUXEDO Aquaris.

i GeForce RTX 4090, Core i9-13900HX and a mechanical keyboard with Cherry MX switches.

Interstellar performance in a compact form factor! **TUXEDO Stellaris 16 - Gen5**



TUXEDO

 tuxedocomputers.com