Zero Trust | grommunio | Storage

ADMIN
Network & Security

ISSUE 83

STORAGE

Developments, management, and new tech

Deduplication on Windows Server 2022

grommunio
Drop-in replacement for Microsoft Exchange

Zero Trust
Planning and implementation

Networking Linux on Azure

Azure Files and File Sync
Classic file shares for clients in the cloud

Rclone
Integrate remote cloud storage

CouchDB
Manage status messages with MapReduce

Dradis
Automated health checks

LINUX NEW MEDIA
The Pulse of Open Source

WWW.ADMIN-MAGAZINE.COM

A Thousand Words Paint a Picture

The thrill and the threat of enterprise automation.

Every system administrator wants to automate the tedious and mundane tasks they must perform regularly. I love to automate the creation of HTML reports. I get a thrill out of creating timed scripts that fire off one after the other to perform elaborate tasks, such as moving or copying files, restarting services, downloading updates, performing system cleanup, archiving files, and so on. There's just something fun about putting things in motion and watching them work without intervention behind the scenes while I research new services, scan performance data, and worry about how much disk space everyone consumes. But there's a dark side to automation, too.

Is it just me, or do we struggle with automation? As you deduced from the paragraph above, I want to automate tasks, but I'm convinced there's a conspiracy to have me automate myself out of a job. I have visions of CXOs sitting around sipping 25-year-old Scotch, toasting, and discussing how they'll cast lots for my salary after I've trained our systems to care for themselves – at least to the point where some AI bot can handle, break, or fix things when they happen. Surely this scenario is my imagination getting the better of me, right? However, losing my job to Lisa, the virtual system administrator, is not the darkest aspect of automation. No, seriously, it isn't.

To me, the worst possible issue with automation is that, when something goes wrong – and it will – it's going to require someone to fix the problem. A human someone. Someone who may or may not be familiar with the systems, the intricate automation, or the problem itself. Think about it.

When you receive a call to fix several dozen systems that no longer deliver their promised functionality, where do you begin? Of course, you'll ask for the documentation. Documentation. That's a joke, right? You've met other system administrators, haven't you? You know that what little documentation exists will not be very helpful. Without detailed documentation, you'll have to unravel this mess on your own, while those who have a financial stake in whatever these systems usually deliver look at you with skepticism when you tell them, honestly, that this will take some time to figure out.

> *If you haven't already begun to try to solve this dilemma while reading this account, you will start by disabling cron or Task Scheduler to stop any automated tasks that run from these system-level timer applications.*

Automation is great when it works, but it can become a nightmare when one thing in the automation chain fails – and many things can go wrong. I'm not bashing automation, but be aware that the more you automate, especially without superb documentation (which you will likely never have or produce), the less you or others will know how every aspect of an automated system works. Trust me. Six months after implementing your own fancy scripts to process file transfers, updates, downloads, and service restarts, you'll feel like you're reading encrypted Martian pig Latin. And that's if they're your scripts. If they're someone else's or you walk into the situation described earlier, you might find the unraveling process overwhelming.

The key to automation is documentation. I hope that's obvious from my description of what happens when things go haywire. However, don't simply add documentation into the scripts themselves; you need to create a guidebook for your automation scheme. A visual diagram of how everything works together is also a good idea.

In one such personal scenario, I tried to alleviate a cascading failure of my automated system by calling scripts from other scripts so that if the first script failed, the second, third, and so on in the chain would never run. This would make the system more supportable, manageable, and easier to troubleshoot because only the first script was called by cron.

My scripts were sufficiently documented, but my guidebook was essential to help anyone understand all the steps in the process.

You've heard the idea, recalled by Spider-Man Peter Parker, that "With great power comes great responsibility." Automation is power, but it implies the responsibility to provide documentation for your system that performs flawlessly until something goes awry. When you get hit by a bus, and your perfect system hits a snag, ensure that someone else can step in and fix the problem quickly and with as few headaches as possible.

Ken Hess • Senior ADMIN Editor

Lead Image © rouslan, 123RF.com

Features

Service

Tools

Containers and Virtualization

Security

Highlights

Management

Nuts and Bolts

@adminmagazine

@adminmag

ADMIN magazine

@adminmagazine

On the DVD

TrueNAS SCALE 24.04 "Dragonfish"
TrueNAS SCALE is a Debian-based, open source infrastructure solution with unified storage that scales up or out. SCALE is easily customized and expanded with the help of integrated apps, Linux containers, and VMs. The basis for all deployments is OpenZFS with its excellent data management capabilities. To scale up, add drives or deploy scale-out object storage by adding multiple systems to a cluster. TrueNAS SCALE means:

- Scale-out ZFS
- Converged compute and storage
- Active-active reliability
- Linux containers
- Easy setup and management

News for Admins

Tech News

Open Source AGPL Added as License Option for Elasticsearch

Elastic has announced that they are adding the OSI-approved AGPL as a license option for Elasticsearch.

"We never stopped believing and behaving like an open source community after we changed the license. But being able to use the term open source, by using AGPL, an OSI-approved license, removes any questions, or FUD, people might have," says Elastic Founder and CEO Shay Banon in the announcement.

Banon says the company chose AGPL because it is a widely adopted, OSI-approved license. Additionally, he says, "We chose AGPL because we believe it's the best way to start to pave a path, with OSI, towards more open source in the world, not less."

Read more at Elastic (https://www.elastic.co/blog/elasticsearch-is-open-source-again).

Sovereign Tech Fund Invests in FreeBSD Development

The FreeBSD Foundation announced that Germany's Sovereign Tech Fund (STF) (https://www.sovereigntechfund.de/) is investing EUR686,400 (around $750,000) in the FreeBSD project "to drive improvements in infrastructure, security, regulatory compliance, and developer experience."

"We are deeply grateful for this significant investment from the Sovereign Tech Fund, which will further enhance security and infrastructure for FreeBSD developers and users," says Deb Goodkin, Executive Director of the FreeBSD Foundation.

The announcement (https://freebsdfoundation.org/blog/sovereign-tech-fund-to-invest-e686400-in-freebsd-infrastructure-modernization/) states that supported work will begin in August 2024 and continue through 2025, focusing on the following areas:

- Zero Trust builds
- CI/CD automation
- Reduced technical debt
- Security controls
- SBOM improvements

Read more at FreeBSD Foundation (https://freebsdfoundation.org/).

Red Hat's OpenStack Services on OpenShift Now Generally Available

Red Hat has announced the general availability of OpenStack Services on OpenShift (https://www.redhat.com/en/technologies/cloud-computing/openstack-services-on-openshift), which is the next major release of Red Hat's OpenStack Platform.

According to the company, Red Hat OpenStack Services on OpenShift helps organizations "manage complexity for faster, simplified deployments of both virtualized and cloud-native applications from the core to the edge all in one place."

Features include:

- Faster deployment of compute nodes.

Lead Image © vlastas, 123RF.com

- Greater flexibility to run applications that are bare-metal, virtualized, and containerized from one platform.
- A scalable OpenStack control plane that can manage Kubernetes-native pods running on Red Hat OpenShift.
- Enhanced observability features.
- Improved security and compliance scanning of the control plane.

Read more at Red Hat (https://www.redhat.com/en/technologies/cloud-computing/openstack-services-on-openshift).

Juniper Networks Offers New AI-Native Courses and Services

Juniper Networks has announced a new Blueprint for AI-Native Acceleration (https://www.juniper.net/us/en/solutions/blueprint-for-ai-native-acceleration.html) to "streamline and accelerate each stage of adoption of the company's AI-Native Networking Platform (https://www.juniper.net/us/en/ai-native-networking-platform.html)."

The comprehensive framework includes:

- Free education to ramp up knowledge and skills.
- Trial offers to let you test the benefits of solutions.
- Flexible licensing.
- Innovative support services to speed deployment.

For example, the company offers a free "AI in Networking for Business Leaders" (https://www.juniper.net/us/en/services/services-for-campus-and-branch.html) course, along with "a range of hands-on classes and actionable certifications (https://www.juniper.net/us/en/solutions/blueprint-for-ai-native-acceleration/learning-and-certification.html) that help IT practitioners plan and start their AI-Native Networking Platform journey."

Juniper also announced new AI-Native Services for Campus and Branch (https://www.juniper.net/us/en/services/services-for-campus-and-branch.html), offering turnkey services to help simplify your network AIOps transition.

Learn more at Juniper Networks (https://www.juniper.net/us/en.html).

Delphix Report Cites Growing Concerns Over Data Protection

Perforce Software recently released findings from the Delphix 2024 State of Data Compliance and Security Report (https://www.delphix.com/report/state-of-data-compliance-and-security), providing insights into the handling of sensitive data in non-production environments, such as development, testing, analytics, and AI/ML.

The recent explosive growth of data has led to increased compliance and security issues, with 91 percent of survey respondents citing concerns about the expanded exposure footprint.

"Our goal with this report is to share the realities of sensitive data exposures in non-production to help enterprises better protect their data moving forward," said Ann Rosen, Director of Product Marketing for Delphix by Perforce.

The top concern cited by respondents involves data breaches and data theft (88%), the report states. Other concerns include:

- Regulatory compliance (86%)
- Ransomware (86%)
- Data corruption and alteration (82%)
- Audit issues and failures (82%)

Additionally, 54 percent of respondents reported having experienced data breaches and theft in non-production environments, while 53 percent reported data corruption and alteration, and 52 percent cited audit issues and failures.

Read more at Delphix (https://www.delphix.com/report/state-of-data-compliance-and-security).

Endor Labs Launches Magic Patches and Upgrade Analysis Tool

Endor Labs launched two new security-related features at the recent Black Hat conference (https://www.blackhat.com/us-24/) in Las Vegas.

The Upgrade Impact Analysis and Endor Magic Patches offerings aim to take the hassle out of hard-to-perform upgrades while also mitigating security vulnerabilities.

Upgrade Impact Analysis shows you what breaking changes a fix could cause to help you more fully understand the impact of a dependency upgrade. Then, if the upgrade is too risky or complex at the time, you can use Endor Magic Patches "to stay safe with a minimal patch that reduces the changes down to just what's required to eliminate the vulnerability," the announcement says (https://www.endorlabs.com/learn/introducing-upgrades-remediation-give-developers-the-confidence-to-fix).

Endor Magic Patches can help you:

- Respond to emerging threats
- Reduce the urgency of upgrading
- Support FedRAMP compliance

Learn more at Endor Labs (https://www.endorlabs.com/).

Rackspace to Offer TuxCare's Extended Linux System Support

Rackspace Technology has committed to delivering TuxCare Extended Lifecycle Support (ELS) (https://tuxcare.com/extended-lifecycle-support/) to its customer base, according to a recent announcement.

Specifically, Rackspace (https://www.rackspace.com/) will deliver TuxCare ELS for CentOS 6, 7, and 8 and Ubuntu 16 and 18. The agreement also includes TuxCare ELS for Hypertext Preprocessor.

"As various Linux operating systems near their end-of-life dates, our collaboration with TuxCare ensures ongoing support and security for our customers," said Jason Henderson, Compute & OS Product Manager at Rackspace.

Read more at TuxCare (https://tuxcare.com/blog/rackspace-technology-to-offer-tuxcare-extended-lifecycle-support-services/).

Announcing eLxr: Enterprise-Grade Linux for Edge-to-Cloud Deployments

The eLxr (https://elxr.org/) project has announced the first release of its open source, enterprise-grade Linux distribution for near-edge networks and workloads.

According to the project website, eLxr provides a secure and stable edge distribution, with a predictable release and update cadence that ensures its suitability for long life cycles and long-term deployments.

In the context of edge deployments, demands such as over-the-air (OTA) updates, data aggregation, edge processing, predictive maintenance, and machine learning features, "necessitate a different architectural approach for both near-edge devices and servers," the announcement states (https://elxr.org/post/elxr-announcement/). These requirements can result in the use of multiple distributions, creating a complex, heterogeneous landscape.

eLxr aims to provide a more homogeneous solution that:

- Caters to applications with stringent timing requirements.
- Relies on a smaller footprint for better performance, optimized workloads, and smaller attack surface.
- Includes built-in security features and dedicated hardware features that include secure boot, Trusted Platform Module (TPM), cryptographic engine, and more.

"With eLxr, the power and stability of Debian and its community serve as the foundation for edge-to-cloud deployments, delivering an enterprise-grade Linux distribution tailored for non-traditional use cases. This project unifies the enterprise tech stack, ensuring accessibility and scalability across edge and server projects while fostering innovation in areas such as near-edge networks," says Mark Asselstine, Principal Technologist at Wind River Systems, which contributed the initial eLxr release.

eLxr is an MIT-licensed Debian derivative, and the project's goals of accessibility, innovation, and open source integrity ensure that users benefit from a freely available Linux without proprietary restrictions.

Learn more at eLxr (https://elxr.org/).

NSA Issues Zero Trust Guidance on Automation and Orchestration

The U.S. National Security Agency (NSA) has issued guidance on "Advancing Zero Trust Maturity Throughout the Automation and Orchestration Pillar" (https://media.defense.gov/2024/

Jul/10/2003500250/-1/-1/0/CSI-ZT-AUTOMATION-ORCHESTRATION-PILLAR.PDF), which represents the final pillar of the Department of Defense's zero trust framework.

The latest cybersecurity information sheet highlights the following three areas where automation and orchestration should be put to use:

- To address repetitive, labor-intensive, and predictable tasks for critical functions and access control
- To enhance critical functions
- To coordinate security operations and incident response

The six other zero trust framework pillars are:

- The user pillar (https://executivegov.com/2024/07/nsa-issues-info-sheet-on-final-pillar-of-dod-zero-trust-framework/)
- The devices pillar (https://executivegov.com/2023/10/nsa-guidance-pushes-for-zero-trust-security-in-dod-devices/)
- The network and environment pillar (https://executivegov.com/2024/03/latest-nsa-csi-promotes-network-and-environmental-pillar-of-zero-trust/)
- The data pillar (https://executivegov.com/2024/04/nsa-releases-guidance-for-advancing-zero-trust-maturity-throughout-the-data-pillar/)
- The application and workload pillar (https://executivegov.com/2024/05/nsa-issues-guidance-for-maturing-application-workload-capabilities-under-zero-trust-dave-luber-quoted/)
- The visibility and analytics pillar (https://executivegov.com/2024/05/nsa-releases-guidance-on-integrating-zero-trust-models-visibility-and-analytics-pillar/)

According to the announcement, this document also offers "recommendations for automating routine tasks to better focus resources on investigating anomalies associated with advanced tactics, techniques, and procedures."

Read more at the NSA (http://nsa.gov/Press-Room/Press-Releases-Statements/Press-Release-View/Article/3833594/nsas-final-zero-trust-pillar-report-outlines-how-to-achieve-faster-threat-respo/).

IT Pros Report Lack of Familiarity with Secure Software Development

A new report from OpenSSF and the Linux Foundation indicates that many IT professionals are not familiar with secure software development concepts and practices.

According to the Secure Software Development Education 2024 Survey (https://www.linuxfoundation.org/research/software-security-education-study), professionals in the following key roles reported being unfamiliar with secure software development:

- System operations (39%)
- Open source program office (OSPO) members (38%)
- Software developers (27%)
- Open source maintainers (23%)
- Security team members (16%)

The lack of familiarity in system operations and OSPO members is concerning, the report says, "as these roles are critical in managing and maintaining software infrastructure and open source initiatives, both of which are fundamental to a company's overall security posture."

Other findings from the report provide insight as to the importance of secure software development training and how tech professionals can acquire it. For example:

- 50 percent of professionals identify a lack of training as a major challenge for implementing secure software development, increasing to 73 percent among data science roles.
- 69 percent rely on on-the-job experience as a learning resource for secure software development, but it can take more than 5 years of such experience to achieve familiarity.
- 53 percent of professionals, especially those in system operations (72 percent), have not taken a course on secure software development, often due to the lack of awareness about good courses (44 percent).

The OpenSSF itself offers training courses — including Secure Software Development Fundamentals — and part of the motivation behind this survey was to identify topics for future courses. As a result of these findings, the OpenSSF says they will focus on developing a new security architecture course.

Read the complete report at the Linux Foundation (https://www.linuxfoundation.org/research/software-security-education-study).

Six storage drivers for Kubernetes

All-Occasion Drivers

In small and medium-sized Kubernetes setups, the choice of the proper container storage infrastructure driver is crucial for the environment's functionality and performance. We look at the most important drivers and their features. By Andreas Stolzenberger

Originally, container platforms ran without persistent storage, but with users increasingly porting stateful applications to stateless container clusters, a solution for persistent storage was needed so containers could save data in a directory before crashing or stopping. This problem prompted the Container Storage Infrastructure (CSI) in 2019 to standardize a series of APIs that Kubernetes can use to request, modify, delete, and connect storage to one or multiple pods.

Of course, storage vendors also provide CSI drivers that convert Kubernetes CSI API calls for the associated storage. CSI drivers control both legacy hardware storage systems and software-defined storage (SDS). In this article, I look at the most important open source CSI drivers suitable for small and medium-sized Kubernetes installations and specify their areas of application. A complete overview of all currently available Kubernetes CSI drivers is available online [1].

CSI Drivers

A CSI driver that dynamically creates storage to match incoming persistent volume claims (PVCs) must be capable of basic functionality, such as creating or deleting volumes. The driver also must address a number of optional functions (e.g., creating snapshots, clones, and expansions) and different forms of access – for example, Read/Write Single pod (RWS) and Read/Write Multiple pods (RWM), where multiple PODs have simultaneous read/write access to a PV. CSI drivers work with either block or file back ends. In the container itself, both simply appear to be linked subdirectories.

Block storage volumes use a local filesystem such as XFS or ext4, which the Kubernetes node manages and `loop` mounts in the container. Block storage PVs are faster, especially for write access, because they use the filesystem cache of the local node but do not support read/write-many access. For the filesystem-based storage classes – Network File System (NFS), Server Message Block (SMB), or CephFS – on the other hand, it is exactly the other way around. RWM mode works, but without a write cache, which must not exist in a network-based filesystem.

A Kubernetes cluster is not limited to a single CSI driver. You have the option of using several storage connections at the same time, and each driver is assigned a storage class. If an application creates a Persistent Volume Claim (PVC), it can be given the name of the storage class. As a rule, every Kubernetes cluster has a storage class annotated as

Lead Image © webreporter, photocase.com

```
storageclass.kubernetes.io/⮐
  is-default-class=true
```

which then acts as the default class and is automatically used for all PVCs that do not explicitly request a specific storage class.

Incidentally, a Kubernetes setup can also work without CSI drivers. In this scenario, the admin either needs to respond manually to an application's PVC and create a PV that matches the claim, or, alternatively, the user can define a series of empty PVs that can then be used by a claim.

Hostpath

Hostpath [2] is the simplest CSI driver for Kubernetes. As the name suggests, this storage class creates subdirectories in a path on the host itself, if required, and then `loop` mounts these subdirectories in the pod that requested the PV. Hostpath offers no redundancy for the PVs it creates, and only works on a single node. However, the GitHub repository comes with a warning for users: *This driver is just a demo implementation and is used for CI testing.*

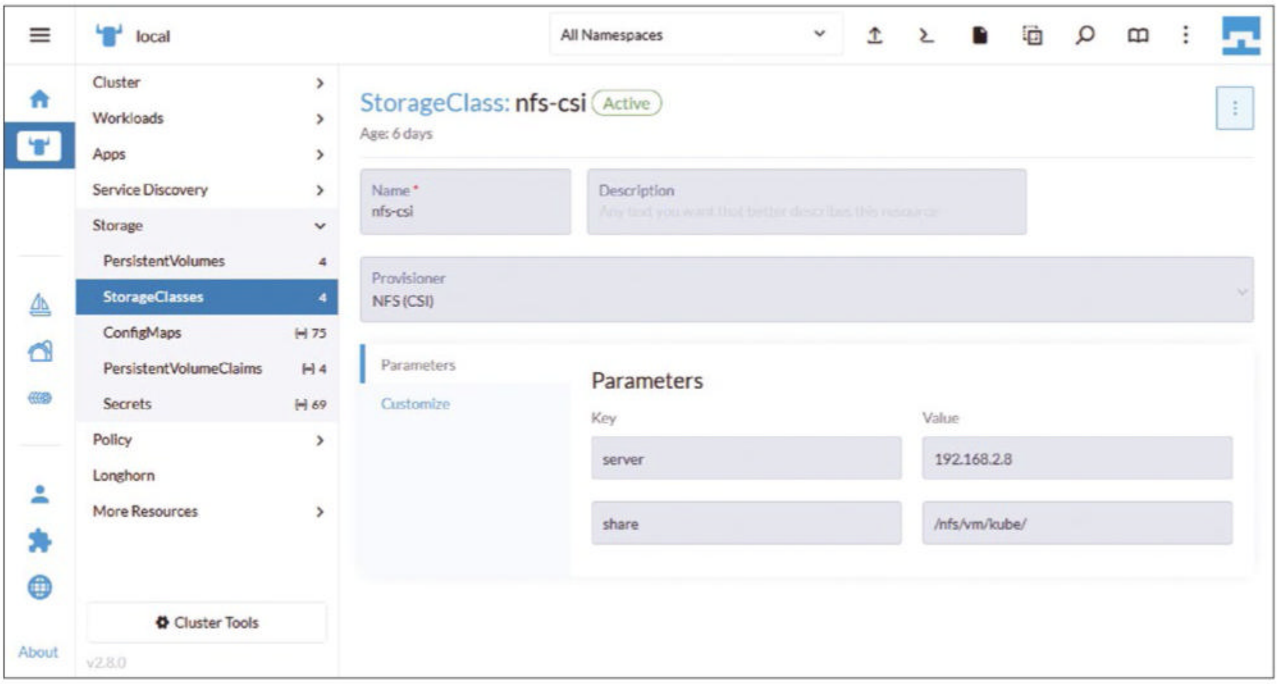
Despite this warning, Hostpath is often used for small single-node setups. Both MicroK8s and K3s use Hostpath in the basic installation. Hostpath can do little more than create and delete file-based PVs. Also, the driver cannot control or limit the memory quota. Nothing stops a pod filling up the entire free disk space on the host by writing to the connected PV. Of course, the CSI driver is very compact, making do with a single container and being frugal in terms of its resource requirements. Thus, Hostpath is suitable for small edge installations, especially on hosts that are low on hardware resources. As described in one of my previous articles [3], edge setups with MicroK8s (Canonical) or K3s (SUSE) run on systems with just 1GB of RAM and one CPU core, even though they have a dynamic CSI driver in the form of Hostpath.

CSI-NFS

The CSI-NFS driver [4] takes the simple strategy of the Hostpath driver to the next level. In principle, this driver works in a similar way by only creating subdirectories for PVs. However, these subdirectories do not reside in an arbitrary host path, but on an NFS server. The host in turn integrates the respective NFS mount into the requesting pod. The driver comprises two components. The first is the NFS controller, which handles CSI tasks such as processing PVCs and creating and deleting volumes; it comes with a snapshot controller, which – as the name suggests – creates snapshots of PVs and can therefore also clone existing PVs. The second component is a CSI-NFS node pod, one per node, that maps the NFS mounts to the respective pods on the node.

As a filesystem driver, CSI-NFS supports `ReadWriteMany` access and can handle dynamically and statically provisioned volumes in parallel (Figure 1). The latter are particularly interesting where systems and applications outside the Kubernetes cluster enable access to NFS resources. Among the various practical scenarios, you can operate a scale-out web server cluster on Kubernetes. You can collect static data such as HTML pages, CSS templates, or multimedia resources on an NFS volume and make the data available to all web server pods. The static HTML data can then be managed by an external system that has the right access to the NFS share.

CSI-NFS volumes are also a good choice as backup drives. Pods and sidecar containers can back up data from running applications and from SQL dumps (e.g., to NFS shares). An external backup or data



StorageClass: nfs-csi Active

Age: 6 days

Name * nfs-csi

Description Any text you want that better describes this resource

Provisioner NFS (CSI)

Parameters

Key	Value
server	192.168.2.8
share	/nfs/vm/kube/

Figure 1: The CSI-NFS driver creates volumes with `ReadWriteMany` access in NFS shares. These volumes are a good choice for backups.

warehouse system can then process the backup data.

CSI-NFS runs independently of the NFS server and supports protocol versions up to version 4.1. The driver supports several storage classes with different configurations in terms of the NFS server, base directory, or protocol version. Therefore, it is also a good choice for virtualized setups (see the "Classic vs. Hyperconverged" box) because it works with existing NFS environments without an additional SDS overheap.

The CSI-SMB driver [5] is an alternative to CSI-NFS, offering a range of functions comparable to those of its NFS counterpart, but using the SMB protocol instead of NFS. This option is interesting in environments with existing Windows file servers.

The GlusterFS [6] distributed cluster filesystem also has a CSI driver. Unfortunately, the project and the associated add-ons seem to have been dormant for a few months. As a commercial provider and the driving force behind the project for many years, Red Hat has largely withdrawn from the GlusterFS project and is set to discontinue support for GlusterFS at the end of 2024.

Classic vs. Hyperconvergent

Many admins face the question of how they can operate their Kubernetes cluster and storage in the most efficient way. Should the Kubernetes nodes run on virtual machines or directly on the hardware? Should legacy storage systems or SDS be used? Do dedicated nodes provide the storage, or do the storage resources come from the nodes themselves in a hyperconverged environment? These questions unfortunately have many possible answers.

If you are considering a complete rebuild of a Kubernetes cluster with your own hardware, a bare-metal installation and hyperconverged storage from the nodes is going to be the most efficient solution, with no need for external, proprietary storage systems with complex storage area network (SAN) connections. Modern hyperconverged SDS delivers better performance and availability at significantly lower costs than classic SAN storage. These setups can then run virtual machines, as well, if required. However, if you set up Kubernetes on an existing virtualization platform with a connected SAN, you need a different infrastructure. SDS with its own redundancy on the virtual machines makes no sense. Given the underlying SAN, which already ensures high availability, a virtual SDS just wastes capacity and performance. You either need very simple CSI storage or the cluster can address the existing SAN storage directly.

TopoLVM

Much like Hostpath, TopoLVM [7] is another single-node storage option – although it is not the whole story.

The driver uses the underlying host's logical volume manager (LVM), managing PVs as logical volumes with, if so desired, a thin-provisioned pool. TopoLVM comprises several components, with the controller managing the storage classes and the incoming PV claims.

The node forwards the controller's requests to the `lvmd` driver and manages the PVs' mount points and filesystems. The Node OS logical volume manager, the `lvmd` service (LVMd), is capable of running as a daemon on the operating system of the Kubernetes node. As a rule, TopoLVM starts `lvmd` as a privileged container in the node pod.

LVMd needs a matching configuration that specifies the volume group names, among other things, to work correctly. On first launch, `lvmd` can import this information from a configuration file; it then saves the information as a Kubernetes configuration map.

Unlike Hostpath, TopoLVM supports quota management. If a PVC requests a volume with 5GB of storage, TopoLVM delivers only a 5GB logical volume, which the connected pod cannot overfill, unlike the scenario with Hostpath. Moreover, TopoLVM can create snapshots to back up PVs – also by LAN to external backup storage.

As mentioned earlier, TopoLVM offers more than single-node storage. The TopoLVM controller can manage several nodes and LVMds and manage PVs that are distributed across several nodes. However, these PVs are then only available to pods that run on the same node as the LVMd. LAN access to PVs is not possible, nor is replication between PVs on several nodes. Therefore, TopoLVM, like Hostpath, is primarily suitable for small single-node setups on edge devices. However, the overheap of TopoLVM and the resource requirements of the driver are greater than with Hostpath. On the upside, the driver does offer quota management and features such as volume snapshots and resizing. TopoLVM is the standard CSI driver in the MicroShift single-node Kubernetes distribution.

OpenEBS

On closer inspection, OpenEBS [8] from DataCore (or, more precisely, from its Perifery division) is not just one CSI driver, but three: Jiva, cStor, and Mayastor. What they all have in common is the use of replication controllers to mirror the nodes' local storage resources on the LAN. The local resources then provide OpenEBS drivers, which work much like Hostpath or TopoLVM.

The three replication engines are aimed at different areas of application. Jiva uses Hostpath as a basis, acting as file-based storage to support read/write-many access, but without snapshots or a write cache. cStor, which relies on a basic driver like TopoLVM to provide snapshot-capable replicated block devices, is faster. Mayastor plays a special role, aimed at low-latency applications with NVMe resources on the nodes.

The OpenEBS low-level CSI drivers can also be used entirely without a replication controller. For example, users could use the OpenEBS Hostpath instead of the CSI Hostpath driver on a single-node setup. The OpenEBS ZFS CSI driver, which creates block PVs in ZFS pools, definitely occupies a special slot for brave admins.

Longhorn

Longhorn [9] is the default storage option in the Rancher Kubernetes distribution. Like OpenEBS, Longhorn relies on a comparatively simple substructure with the addition of LAN replication. Longhorn creates block devices on selected storage nodes not by LVM, but as image files in the regular filesystem of the Kubernetes node. The node CSI driver then uses iSCSI to publish these images to the LAN, where the Longhorn controller can mirror them across several nodes. The volume mount is handled by the Longhorn client, which works on all nodes and establishes the storage connection by iSCSI. As a result, Longhorn either lets dedicated nodes provide the storage resources or uses hyperconverged storage on all nodes. Of course, iSCSI transport for PV access and replication places a burden on the network, as is common with other storage replication technologies. However, Longhorn does not necessarily have to handle storage access over the primary LAN of the Kubernetes cluster on which the applications themselves run. Instead, Kubernetes network plugins can be used to move the storage traffic to a separate LAN.

A Longhorn setup can control several storage classes if required (see Figure 2), which means users can use a three-replica setting for critical applications while relying on unreplicated volumes for test setups where redundancy requirements are not as strict. The latter case is also practical where administrators use single-node Kubernetes setups. When the cluster is expanded, the storage system can also grow, adding further nodes and implementing redundancy.

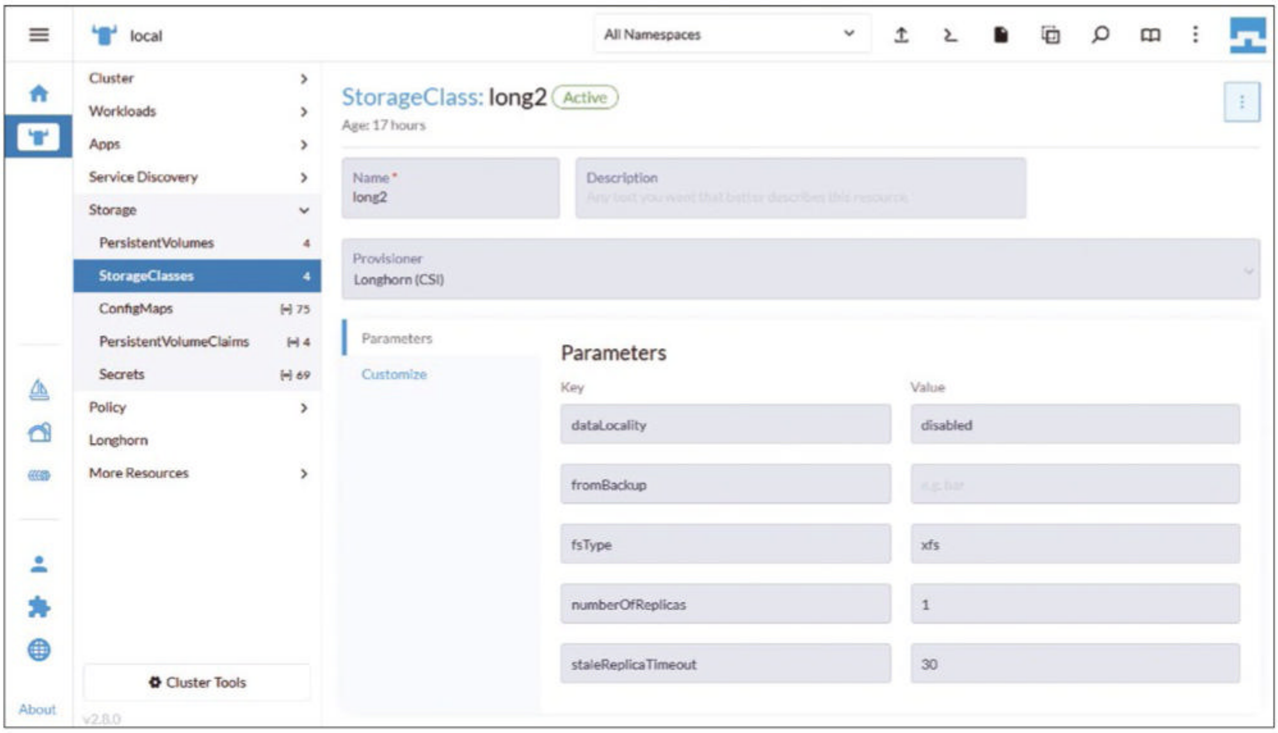
Additionally, nodes can be tagged and then assigned to different storage classes, which means you can bundle nodes with NVMe or solid-state disk storage in a fast pool and assign nodes with legacy disks to a slower storage class with greater capacity. By default, Longhorn uses ext4 to format the block storage, but you can change the storage class definition to XFS.

Longhorn comes with a simple web user interface that primarily gives you an overview of the storage resources used and their replication states.

The storage classes are configured as a YAML declaration. Snapshots of PVs can be created in the graphical user interface, or entire drives can be backed up. All Longhorn needs is an NFS share.

Rook for Ceph

For the sake of completeness, do not forget the Rook operator, which sets up hyperconverged Ceph clusters on Kubernetes. I will not go into the details of the operator and the resulting Kubernetes storage here, because Ceph is not suitable for medium-sized Kubernetes setups and definitely not for small setups.



StorageClass: long2 Active

Age: 17 hours

Name * long2

Description Any text you want that better describes this resource

Provisioner Longhorn (CSI)

Parameters

Key	Value
dataLocality	disabled
fromBackup	e.g. bar
fsType	xfs
numberOfReplicas	1
staleReplicaTimeout	30

Figure 2: **Longhorn supports multiple storage classes with different settings. The *long2* class in this cluster uses the XFS filesystem to create PVs without redundancy (replica 1).**

The resource requirements of the distributed object store are high in terms of CPU, memory, and – above all – network load. Any detailed description of the operator and the configuration of a hyperconverged Ceph cluster is well beyond the scope of this article.

Conclusions

Even if it sounds soberingly simple and old-fashioned, every Kubernetes cluster should have NFS storage as the second CSI driver in the system. This driver can be used to handle backups with or without sidecar containers, to aggregate logs, or to set up RWM storage for clustered applications. That said, the NFS connection should not be used as primary storage because the write performance is not up to the task of handling I/O-intensive jobs in containers. Drivers such as TopoLVM and Hostpath are only suitable for single-node edge setups that you are not planning to scale up. However, this setup is likely to work well on the network edge, especially in combination with an NFS connection for backups.

On the other hand, if you are designing the single-node setup as the starting point for a growing cluster or are starting with a three-node setup from the outset, you will want to use Longhorn or one of the OpenEBS storage systems to set up redundant and high-performance PVs. Longhorn is the approach of choice because it makes setting up and operating hyperconverged storage far simpler than OpenEBS, and you get a simple user interface and the integrated NFS backup feature on top. ■

Info

[1] Available CSI drivers: [https://kubernetes-csi.github.io/docs/drivers.html#production-drivers]

[2] Hostpath: [https://github.com/kubernetes-csi/csi-driver-host-path]

[3] "Simple, Small-Scale Kubernetes Distributions for the Edge" by Andreas Stolzenberger, *ADMIN*, issue 80, 2024, pg. 46, [https://www.admin-magazine.com/Archive/2024/80/Simple-small-scale-Kubernetes-distributions-for-the-edge/]

[4] CSI-NFS: [https://github.com/kubernetes-csi/csi-driver-nfs]

[5] CSI-SMB: [https://github.com/kubernetes-csi/csi-driver-smb]

[6] Gluster CSI: [https://github.com/gluster/gluster-csi-driver]

[7] TopoLVM: [https://github.com/topolvm/topolvm]

[8] OpenEBS: [https://github.com/openebs/openebs]

[9] Longhorn: [https://github.com/longhorn/longhorn]

The Author

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

Celebrate LPI's 25th Anniversary with Free Exam Vouchers!

Thank You for All Your Support!

Linux Professional Institute (LPI) celebrates its 25th anniversary in 2024. To mark this milestone, we are offering something special: From October to December 2024, every 25th candidate who takes one of our exams will receive a free voucher for their next exam. Thank you for supporting LPI! Celebrate 25 years of promoting Linux and open source expertise with us and advance your IT career!

Find out more about
Linux Professional Institute
and our anniversary at lpi.org/25

Linux
Professional
Institute

Developments in storage management

Putting All Your Ducks in a Row

The goal of storage management is to ensure efficient and reliable data usage, both to improve the performance of storage, server, and hybrid IT infrastructures and to ensure their integrity and availability. We look at likely developments in corporate IT as a result of proactive data management. By Norbert Deuschle

Storage management covers the key aspects of enterprise management of storage resources and the components involved, both on premises and in hybrid clouds and multiclouds, including functions for resource provisioning, process automation, load balancing, capacity planning and management, predictive analysis, performance monitoring, data replication, compression, deduplication, snapshots, and cloning. Other tools include services for cloud storage and container management systems such as Kubernetes and the like.

In contrast, storage resource management (SRM) is a sub-aspect of enterprise storage management. From direct-attached to fabric-attached storage, SRM is typically more closely tied to deployed storage hardware, whether in the form of software-defined JBODs (just a bunch of disks) or self-contained intelligent storage arrays and subsystems. SRM also applies to hyperconverged environments and computational storage systems.

Accessing Data

A data-centric approach is increasingly observed in today's application operations and the associated IT environments. Information is used as a strategic resource that needs to be managed, analyzed, and protected to reflect its value. This data landscape typically consists of local environments and cloud systems from different providers, which can be geographically distributed, making data management itself more difficult.

Data and storage management are interdependent disciplines. Hardware-related storage management relates to physical storage resources and media such as hard disk drives (HDDs), solid-state drives (SSDs), tapes, and other devices. Storage-related data management is about the stored data, regardless of the physical storage medium.

In addition to the associated processes, enterprise data management comprises the organization, backup, manipulation, analysis, and management of all relevant data throughout the entire life cycle, including database management, data integration, modeling, and quality assurance. Setting up processes to identify a company's critical information is closely tied to how the storage management resources are prioritized.

For example, email might be an organization's top priority, but storing and archiving email data for a specific group (e.g., chief experience officer, CXO, leadership) will be more critical in the financial sector than other areas, such as marketing. It is important to make sure that these priorities are taken into account on the storage side.

Standardized Interfaces

Enterprise storage systems offer support for common management platform APIs. The Storage Management Initiative Specification (SMI-S) for integrating resources from different providers – to support shared use and boost operational efficiency – has been increasingly adopted by manufacturers since 2021. The aim of standardization from an operations perspective is to manage storage devices from different providers as simply and securely as possible; after all, thanks to SMI-S, they all look the same "from the outside" and behave in an identical way, which in turn enables orchestration.

Version 1.2.5a Swordfish from June 2023 (version 1.2.6 since January 22, 2024) delivers an extension of the DMTF (formerly known as the Distributed Management Task Force)

Photo by DESIGNECOLOGIST on Unsplash

Redfish specification, allowing the same RESTful interface to be used – in combination with JavaScript Object Notation (JSON) and Open Data Protocol (OData) – to manage storage devices and services transparently with block storage, filesystems, object storage, storage network infrastructures, and servers (Figure 1). This use of Swordfish and Redfish configurations and resources also applies to the management of NVMe and NVMe over Fabrics (NVMe-oF).

The ability to connect to the cloud and manage it with Amazon Simple Storage Service (S3) as the de facto standard for object storage is now an option for storage systems virtually everywhere. In addition to automation and orchestration functions, other components of comprehensive (cloud) data and storage management include performance monitoring, security, governance, compliance, and transparent cost management.

Integrative Strategies

Software-defined storage (SDS) products generally offer greater flexibility and scalability than hardware-only variants. In particular, the ability to adapt resources dynamically and independently of the platform, expand them across regions, and extend them into the cloud enables efficient scaling, but also requires more careful planning and implementation. SDS systems are thus more complex to manage because of their inherent flexibility and the multitude of options and settings that depend on the size and heterogeneity of the application. This essential configuration, monitoring, and ongoing system optimization work needs to be factored in on top of the current shortage of skilled workers.

The trend toward convergence of storage management at the infrastructure level with storage-related data management tools is not affected. Organizations that take a more integrated approach to storage and data management are able to accelerate application delivery and get rid of infrastructure silos. From an administrative perspective, software-defined infrastructures (SDIs) have the advantage that two objectives can be kept in mind in terms of integrative IT: mobility and elasticity of the storage layer during operation. More capacity or computing power means adding these resources on the fly.

SDI uses cost-effective, scalable resources in the public cloud, which enables organizations to adapt to changing requirements in a better way and

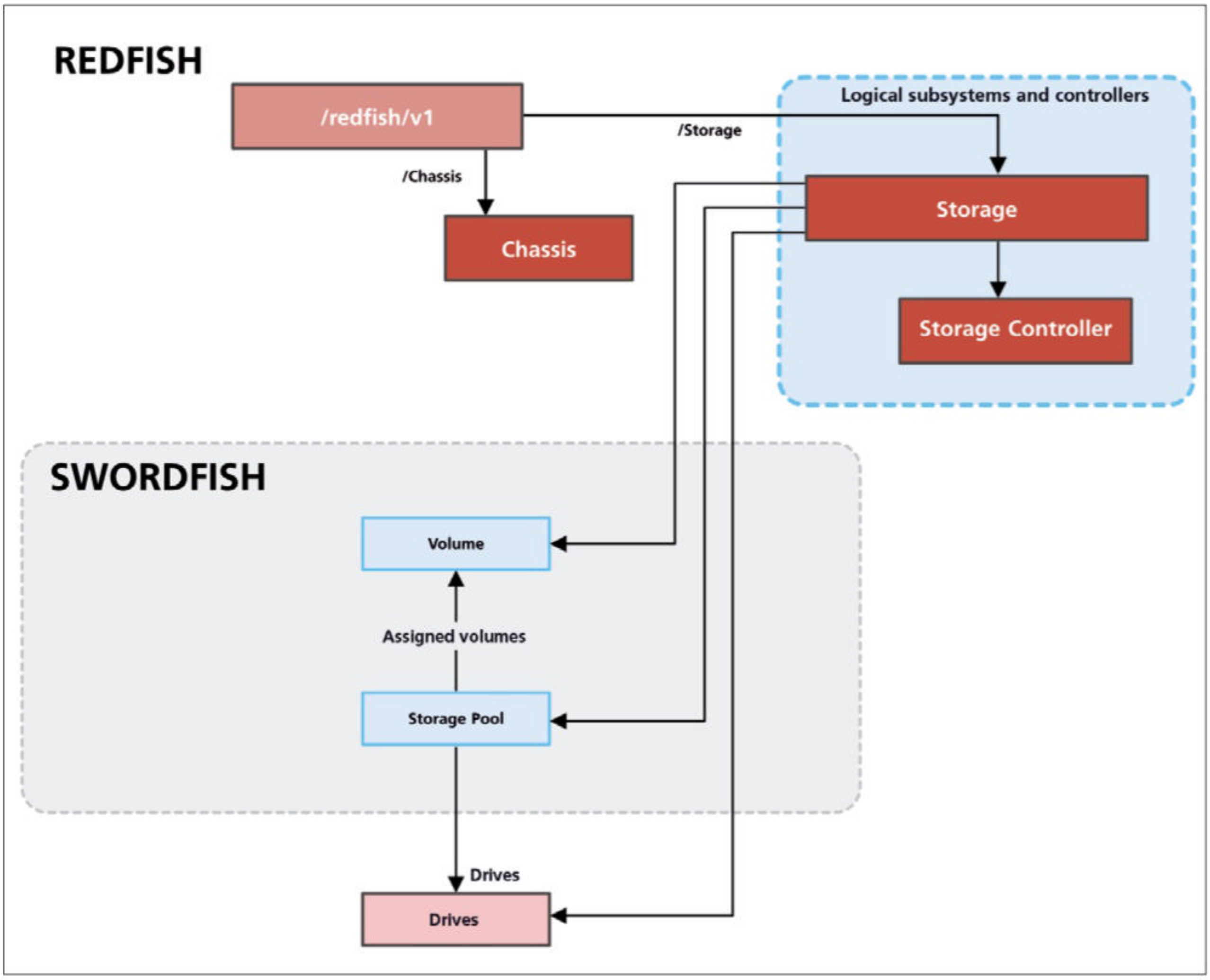


Figure 1: Swordfish standalone configuration with v1.2.5a of the Storage Management API specification.

respond more quickly. In technical terms, SDS separates the control plane of the storage infrastructure from the purely physical hardware. It supports centralized management of the various storage resources and the automation of the processes involved through storage and data management functions such as deduplication, compression, snapshots, replication, and data encryption. Depending on the provider, this means that critical data can be managed and backed up fully "in the box" (i.e., without resorting to additional external toolsets).

Thanks to software-controlled enterprise storage, both the on-premises storage and the native public cloud storage environment can now be managed as an integrated SDI – as if the public cloud were just another storage system identified by its user interface. The public cloud is typically used for storage-intensive use cases such as data archiving – also for backups, disaster recovery, or cloud-native DevOps. What this combination gives organizations above all is better control because critical data can be kept on site to ensure legal compliance with data governance, compliance, and data protection regulations.

AI Support for SDIs

A data services infrastructure that supports hybrid (multi)cloud operation is a key element of any modern data management strategy. The data management and storage functionalities for rapidly growing distributed file and object storage systems are logically combined on a management platform to cover the widest possible range of use cases for unstructured data, while reducing the costs of storage and data management.

This infrastructure also has security benefits. With the help of artificial intelligence for IT operations (AIOps), a highly automated infrastructure is in a far better place when it comes to acting as a resilient platform for managing data services. The use of generative artificial intelligence (AI) and natural language processing also opens up further opportunities to accelerate cyber recovery with the help of AI functions to meet service level agreement (SLA), recovery time objective (RTO), and recovery point objective (RPO) targets more effectively. One example is the collaboration of backup platform providers with Microsoft Sentinel and Azure OpenAI Service. The use of AI significantly reduces the time required to investigate and define responses to cyber incidents. Armed with proactive ransomware detection or critical data identification, modern AI-powered backups are able to use machine-learning (ML) technologies to minimize the potential risks and support threat detection (e.g., by providing immediate warnings if data is encrypted, unauthorized changes are made, or other anomalies occur).

In this way, the introduction of AI and ML transforms the backup and recovery processes – which were, to a great extent, previously reliant on manual intervention – into a highly automated, data-driven process that enables more consistent coordination with IT service management tools. These tools establish a central interface between IT and corporate management, which is increasingly forced to assume greater responsibility for the availability and security of applications and systems under the pressure of growing regulatory requirements. External catalysts and drivers of these developments relate to European Union (EU)-wide political and regulatory requirements such as information and communications technology (ICT) resilience and measures to improve cybersecurity, including the current implementation of the EU Digital Operational Resilience Act (DORA).

DevOps Challenges

In practice, however, storage and data management integration in the sense of a consistent overarching architecture between on-premises and native cloud resources still poses various challenges, especially in the DevOps environment – not least because the number of stateful applications has increased very rapidly and, in some cases, uncontrollably in recent years. Persistent storage systems also run on separate systems outside native cloud environments, which makes the management overhead more complicated, less clear, and more expensive.

Interest in the provision of stateful applications with Kubernetes cluster systems has also increased because organizations do not always want to, or are not always able to, rely on a separate team to manage the storage. Provided you have the necessary expertise, having the IT team that manages the Kubernetes cluster also manage all the storage resources connected to this cluster will probably be more cost effective, while saving time and facilitating operations in nearly all cases. As a result, the need is growing for increased automation of storage management across all types of clusters to which different storage services can be connected. Developments in the field of AI-supported toolsets promise a remedy.

Tools are now available on the market to enable shared storage-related data management within combined file and object platforms – either in the form of cloud-native software or as an appliance that combines hardware and software.

Docking Containers with CSI

Strong growth in the area of unstructured files and the resulting increasing attractiveness of object storage for new applications underlines the need for comprehensive consolidated storage management that is essentially based on software-defined compute and memory resources. In SDIs, SDS uses a distributed system architecture for primary, secondary, and archive data that goes hand in hand with the development trend of breaking down isolated infrastructure silos in favor of flexible and scalable cloud-supported hybrid platforms.

The automation of DevOps processes to accelerate the process of creating cloud-native and scalable applications on the basis of containers and microservices plays an important role. The cloud-native ecosystem has offered specifications for storage management for some time, in the form of the Container Storage Interface (CSI) standard, to achieve a standardized, portable approach to the implementation and use of storage services that are based on containerized workloads in a faster and easier way. By definition, CSI is another option for providing arbitrary block and file storage systems for containerized workloads.

CSIs can be used to interact with storage on a local server or outside the cluster itself. CSIs are also available for interaction with Azure or S3 storage to support the use of cloud-native platforms for storing pod data. The data persists even if a pod itself disappears. Vendors such as NetApp, Pure Storage, IBM, HPE, VMware vSAN, and others offer storage plugins for Kubernetes and the like. On the tool side, open source projects such as Kubestr are capable of analyzing the relative performance values of various storage configurations, even across cloud providers.

Efficient Data Management

Modern cloud-native applications that are based on containers and microservices integrate scalable file, block, and object storage services directly in the application cluster and merge them with other applications and services that allocate the storage. This combination makes the cloud-native cluster self-sufficient and portable across public clouds and on-premises deployments, which in turn helps organizations modernize their data centers with dynamic application orchestration for distributed storage systems in local and public cloud environments.

However, even where higher level management tools such as Kubernetes use distributed filesystems such as NFS and GlusterFS, the use of a container-enabled storage fabric is still recommended if the design brief involves meeting the requirements of stateful workloads in production. Operators can choose from a growing number of open source projects and vendor platforms, such as OpenEBS, Rook Ceph, Longhorn, GlusterFS, or LINSTOR, and from a range of commercial implementations by SUSE or Red Hat, to name just two.

As in legacy production operations, the increasing requirements in terms of high availability, performance, compliance, data security, and cyber protection must be taken into account in this environment. The key performance features when selecting a system are:

- Storage as a service to be capable of providing the services in cloud style.
- A high degree of automation, so that provisioning does not have too close a tie to IT.
- Optimized integration with existing developer processes.
- End-to-end functionality (edge-to-core to multicloud).

A growing number of storage-as-a-service offerings have been observed – in the past year in particular – that aim to facilitate the provisioning and management of server and storage resources. The motivation behind these offerings is often to limit capital expenditure (CapEx) and to simplify procurement, deployment, and ongoing maintenance, especially where investments are typically not fully used during operations. Cloud offerings for IT management services can specifically help rationalize the entire infrastructure operation in the storage environment. Examples include backup, disaster recovery, or archiving as a service.

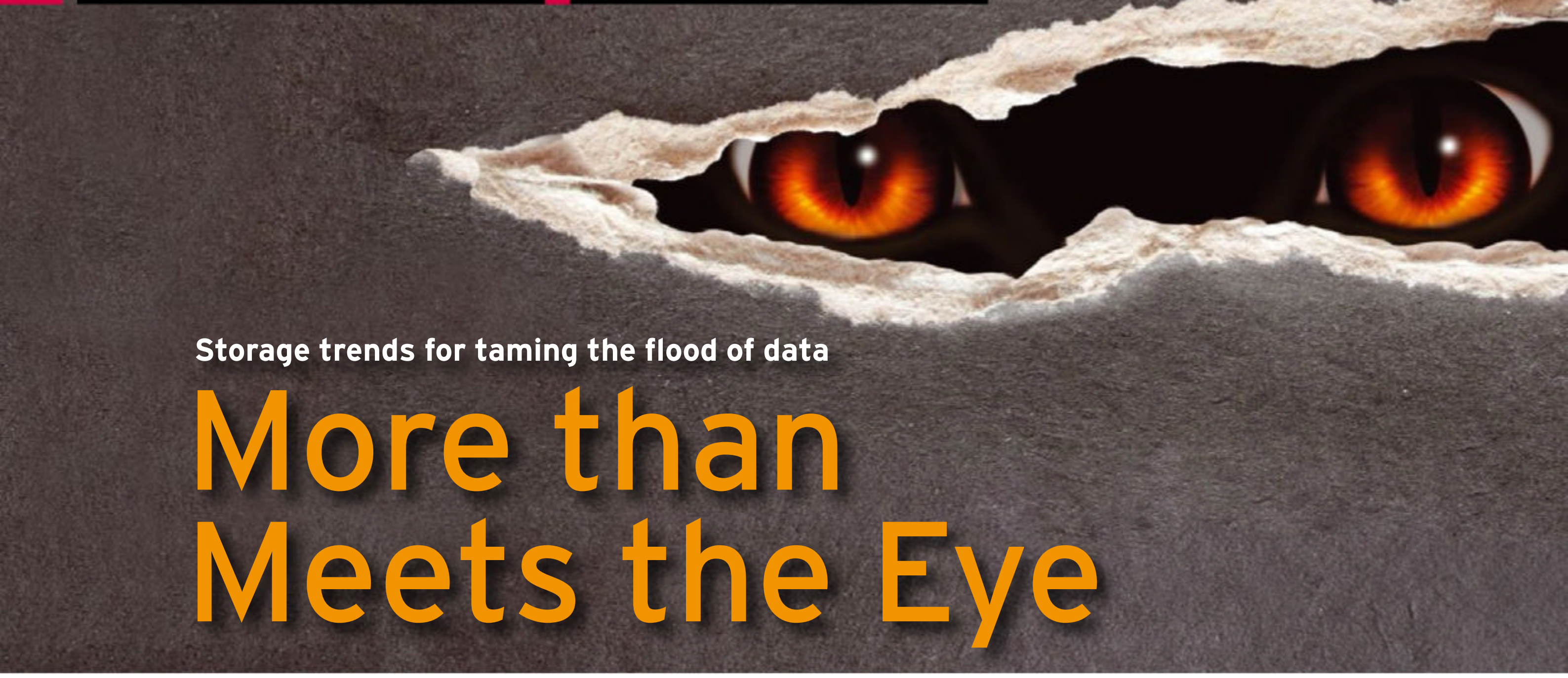
These offerings also apply to virtual desktops with data as a service (DaaS), which simplifies desktop provisioning and therefore data storage management. That said, operators must ensure that their virtual desktop infrastructure (VDI) has sufficient storage resources to meet the constantly growing performance and capacity requirements. In the case of DaaS offerings, this usually translates to additional charges. In the case of on-site VDI, this means organizing more storage space and boosting performance with faster all-flash arrays in response to greater service acceptance.

In general, subscription services or on-demand services will only be successful in the future if they are guaranteed by matching (i.e., reliable and measurable) SLAs, particularly in the fields of data protection, energy efficiency, and sustainability.

Conclusions

Today, IT organizations no longer simply manage storage resources: They also view data management as a unique selling point in interactions with customers, competitors, markets, and applications. Storage management tools not only make it easier to monitor, manage, and troubleshoot all the key components of a storage system but also control important parameters for compression, deduplication, and performance optimization. Modern platforms also provide advanced functions for setting up a persistent cloud-native environment with a high degree of automation and SLA functionality for application data – without having to tie provisioning to IT operations.

Containers, service networks, microservices, immutable infrastructures, and declarative APIs are required to ensure application performance for hybrid cloud environments. In combination with robust AI-supported automation, they together enable the implementation, monitoring, and management of loosely coupled systems, which are then more resilient, relatively easy to handle, and centrally manageable. In terms of sustainability, energy-efficient platform use for data centers will be a key factor in reducing CO^2 emissions. Organizations will need to take this into account in both their purchasing decisions and their planning. ■



Storage trends for taming the flood of data

More than Meets the Eye

The Open Compute Project, Microsoft's data storage on glass, and standardized protocol structures for the IoT era are pioneering open storage technologies for future-proof hardware and software that make storage systems more durable, more manageable, and easier to repair. By Ariane Rüdiger

The unabated growth of data that requires higher density data carriers has made storage one of the most exciting IT topics. Hard disks and solid-state disks (SSDs) have to be replaced every five years or so, creating mountains of electronic waste. Most of these components end up in the shredder for data protection reasons. Microsoft, for example, claims that it scraps millions of hard drives every year, which is not compatible with the company's ambitious environmental goals of offsetting all carbon emissions completely in just a few years. This goal is unlikely to occur without improved storage. The proposed solution is Microsoft glass. Another solution proposed by the Italian protocol manufacturer ZettaScale is the Zenoh data protocol.

Old Tech

Tape, although considerably more durable than hard disks, SSDs, and flash drives, doesn't really have much of an effect for long-term storage – at least not when compared with truly long-term approaches to storing knowledge such as books, microfiche, or (as an extreme example) hieroglyphics, which are legible for thousands of years because they are carved in stone or other permanent media. What's more, physical destruction of hard disks no longer offers any guarantee of rendering the data unreadable. On the contrary, it is already possible today to regenerate data even from the tiniest hard disk remnants. Therefore, improved methods or more secure storage media are urgently needed.

Another problem on the protocol side is the large numbers of what can be high-volume, roundabout requests and data transports required for applications such as industrial Internet of Things (IoT) or autonomous transport. Because of today's protocol structures, the overall system always needs to know where to find the required data. If the data is elsewhere, the request first passes through many routers to its destination. All told, this generates massive overhead, especially in terms of energy consumption for communication. The more branches the network has, the greater this problem becomes.

Compared with these basic issues, other challenges are more cosmetic. However, better technologies could significantly improve the situation in these cases, too. Just consider, for example, the waste of storage space and energy and the unnecessary effect on service life when writing to SSDs. When you transfer 1MB to an SSD, the drives actually writes considerably more data. This effect is known as write amplification (WA), which occurs because SSDs erase far more roughly than they write. As a result, more data is moved and rewritten during rewriting than the storage medium has received (garbage collection).

WA is measured as the write amplification factor (WAF), the ratio of the data written compared with the data sent to the storage system. If you send 1MB of data to an SSD, but the SSD writes 1.5MB, the SSD has a WAF value of 1.5. A WAF of 1 with zero overhead would be ideal. Sequential write operations come closest because they store large contiguous blocks of data. The WAF is highest when small snippets of data are written randomly, which is becoming increasingly common (e.g., because of the growing number of IoT environments).

Error diagnostics and tests for certifying storage media for OpenShift Container Platform (OCP) environments are also complicated and time-consuming, because they need to make it as clear as possible what a storage medium is suitable for and what you can expect from it. However, each hyperscaler uses different tools, some of which might be specific to a manufacturer, leading to gaps in the test suite. Moreover, the existing SSD error correction functions have only ever been accessible to hyperscalers.

Not all behavioral data generated when using the medium reaches the troubleshooter on the user's side because of data protection. For example, you are still unable to access human-readable, manufacturer-specific test or diagnostic data when troubleshooting

Lead Image © Natalia Lukiyanova, 123RF.com

storage media because the binary data must not leave the data center. Thus far, you have been forced to rely on the manufacturer's response. Furthermore, if you have a heterogeneous environment, you are forced to work with many different firmware versions.

Most of these issues expose hyperscalers to pressure as a result of the size of their installations. The protocol problem, on the other hand, stands in the way of autonomous transport and effective industrial IoT. It therefore comes as little surprise that many innovative solution proposals are either hatched within OCP, which was founded by IT major league players, or introduced into the OCP biosphere as soon as they are deemed to be reasonably practicable; however, other efforts are taking place elsewhere, as well.

Flexible Placement

The WA problem was initially addressed by overprovisioning, with host-side load-balancing instructions then being introduced in 2007. In fall 2023, the NVM Express (NVMe) committee adopted a new procedure for optimal data placement on flash memory: flexible data placement (FDP) [1]. FDP is defined in NVMe technical proposal TP4146. The idea is to adopt the technology into the requirements for OCP-certified storage. Essentially, FDP is about the host providing a virtual pointer for where best to store the data in the storage medium. The storage system then only needs to be capable of understanding the instructions, if available, and of placing the data in the prescribed superblock instead of choosing one itself. If no such notification occurs (i.e., the data comes from a system that is not FDP-enabled or the NVMe system itself is not FDP-enabled), it's back to business as usual: write, read, trim (i.e., notifying that specific data on the SSD is no longer required and can be deleted), and all the other security mechanisms.

On the upside, this method ensures backward compatibility with previous storage systems or application versions. Applications will benefit once they support the new process, but they will still run if they don't, and devices can switch FDP on or off.

FDP is already in place in version 5.19 of the Linux kernel, in the cross-platform xNVMe tools starting at version 0.7, in the free Qemu hardware simulator starting at version 8.0, and in `nvme-cli`, the NVMe command line. Further implementations are underway. The process has already been tested in several heterogeneous environments, where the data flowed to the SSDs with the use of different "writers" (applications, containers, virtual machines, namespaces, microservices, write patterns, etc.).

OCP presented the results at a storage conference last year (Figure 1). The results were that FDP reduces the wear and tear on data carriers by half, or even two thirds, and doubles or triples throughput. The WAF is consistently considerably lower, often with even a value of 1 – or close to it. According to OCP, implementation of the new process is progressing rapidly.

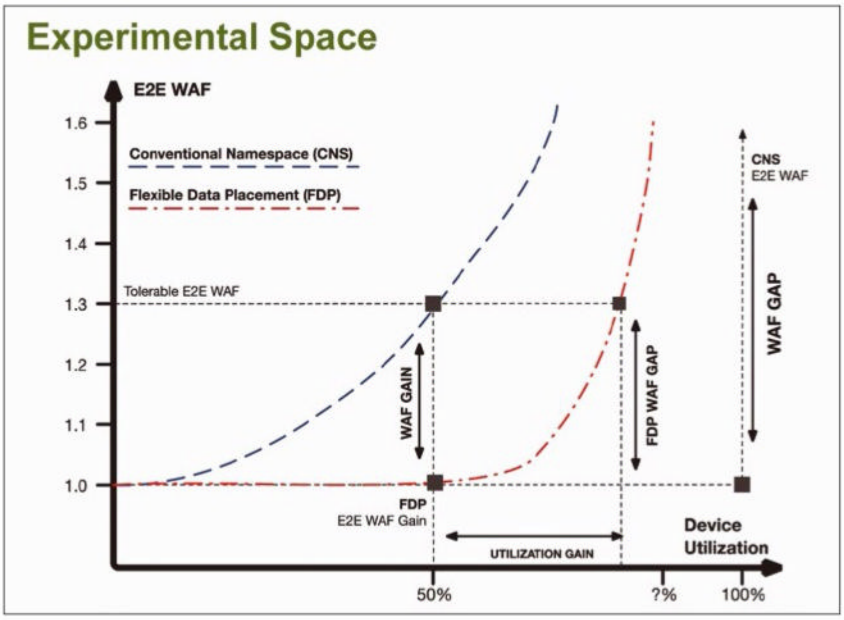


Figure 1: Compared with legacy approaches, FDP fends off undesirable WAF levels up to a higher medium utilization level than previously possible.

More Transparency for Test and Diagnostic Data

OCP is also doing something about the test and diagnostics problems with SMART logs or manufacturer-specific logs to which people had access in the past that were of little practical use. The information the logs contained was too superficial to help people discover and eliminate the cause of error.

For data protection reasons, telemetry logs currently can only be read by the vendor. OCP is trying to improve the situation with several specifications included in the OCP data center NVMe specifications. First is to define a health information log with SSD statistics (version 1.0) suitable for monitoring masses of SSD media. Second, is a latency monitor that isolates current performance peaks to enable real-time debugging (version 2.0). Third is formatted telemetry, nearing completion in version 2.5, that results in human-readable telemetry data, so the user has independence from the storage system vendor when troubleshooting storage media.

OCP provides a simple interface in C++ and Python. The new telemetry software can also be integrated with existing test platforms that work with the vendor-independent OpenTAP, OpenTest, Fava, or ConTest interfaces. OCP has also developed the new Northstar certification procedure.

The idea is to outsource hyperscale qualification of NVMe SSDs to manufacturers or to the test facilities they commission. At the end of the day, other users could also benefit. The project includes a repository for open tests, open test development, and openly available tools. The whole thing is glued together by a clearly defined continuous integration and continuous delivery (CI/CD) process.

Formatted Telemetry in Detail

Now the aforementioned human-readable telemetry data comes into play. OCP's work is advanced in this respect. Up to now, troubleshooting has been based on NVMe telemetry data. Specifically, areas 1 and 2 of the log pages, that fill the storage medium with information, have been redefined starting in version 2.5 of the telemetry log page (telemetry page hereafter) to contain data in a total of four separate data segments. Additionally, a cross-disciplinary page, the OCP String Log page (string page hereafter), is also planned as a kind of directory of all encrypted strings on the telemetry page along with their ASCII translations. Samsung has taken on the task of defining the format.

As for the technical details, I'll cover the structure of the string page first. The header points to the starting addresses of the other areas of the page and four sections. Three of the sections contain identifiers (IDs), and the fourth contains the ASCII strings to which the IDs translate.

The first section contains the IDs for vendor-specific statistics, including sub-sections. Each ID entry comprises the ID to which it points, a field with a length specification, and details of the offset to the start of the respective section. Section 2 contains vendor-specific event IDs. The event fields are structured in the same way as in the statistics section, but with an additional Debug Type field.

The third section contains further vendor-specific event IDs defined by OCP. The entries are structured in the same way as those in the vendor-specific event ID section. Finally, the fourth section contains the human-readable ASCII strings along with the associated IDs. The entries in the first three sections are sorted such that the entry with the shortest ID is assigned a counter value of 0; the longer an ID, the higher the counter.

The telemetry pages access this string page. A telemetry page is divided into a header and four sections (Figure 2). Sections 1 and 2 are particularly important for human-readable diagnostics data, which is why they are the main focus here. Section 1 has a fixed size and must not affect I/O latency during readout. Section 2 is defined by the vendor and can influence I/O latency.

The data in section 1 is broken down further. Each unit contains an OCP header; statistics; and first in, first out (FIFO) event memory. A maximum of 16 named FIFO event buffers are permitted; otherwise, their number is defined by the vendor.

The header defined by OCP for data section 1 contains basic information on the version, profiles, statistics (number of entries, which is vendor-specific), the number of FIFO buffers, the log pages for NVMe SMART and health information, and the extended OCP SMART and health information. In addition to the data, the individual entries in the statistics field each include an ID, information on persistence, a reference namespace, and the data volume. The format supports several profiles that can differ in terms of the data section layout and in terms of the statistics, FIFO buffers, and events that are acquired. The ID tells you whether the statistics were

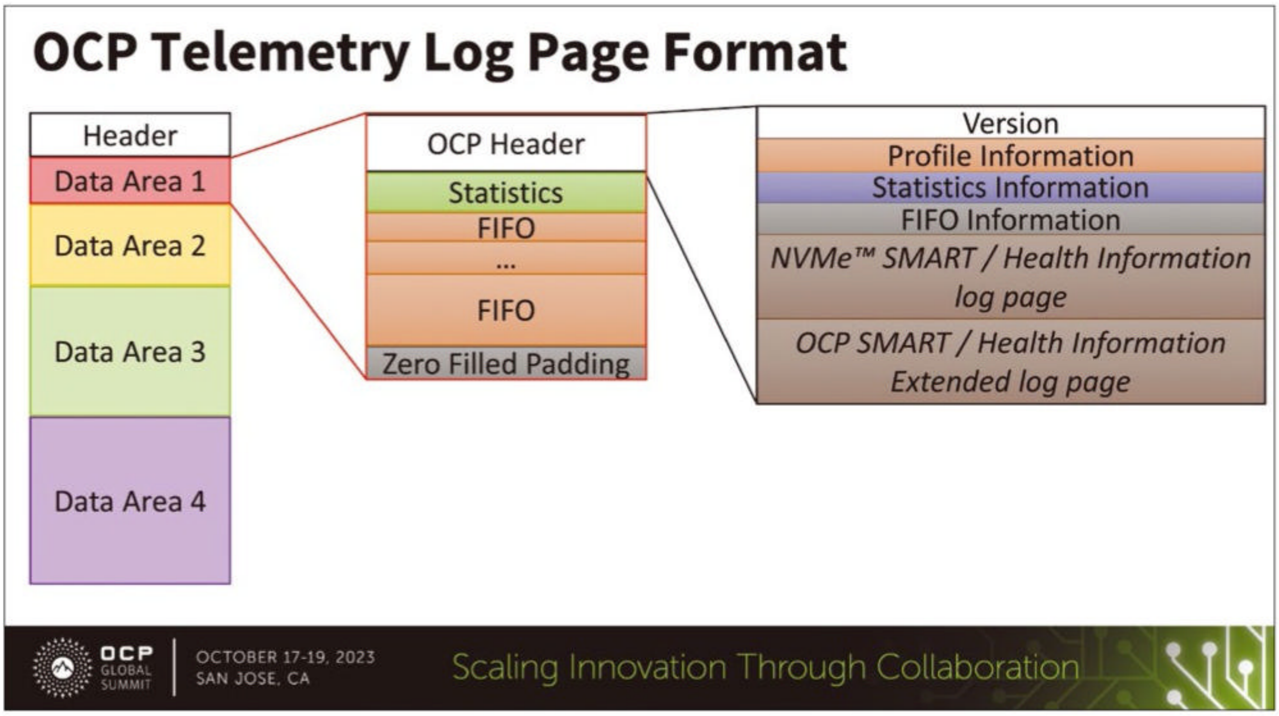


Figure 2: Structure of the OCP telemetry page with the structure of data section 1 and the OCP header.

defined by OCP or by the vendor. The persistence indicator tells you whether the data is retained after a reset or failure. The oldest entry is assigned a serial number of 0.

Each individual entry in a FIFO buffer contains several fields: a debug type (e.g., NVMe, PCIe, reset), a matching ID, the data volume, vendor-independent data, the vendor identification for the specified ID, and vendor-specific data.

The data in data section 2 is structured like the data in field 1, but without the OCP header. Instead, the addresses of the data in area 2 are specified as an offset to the start field of data section 1. From the identifiers, you can now read the data you need from the system in the string page.

This format is close to adoption and looks likely to make the lives of storage troubleshooters in hyperscale environments far easier. One hopes, and it is even likely, that this technology will establish itself in other storage environments.

Glass Storage

Microsoft has taken an in-depth look into the issue of storage medium transience. The result is an advanced strategy for a storage system on glass media. Project Silica [2] was launched by Microsoft around six years ago at the University of Cambridge. In 2022, it presented its initial results at a Microsoft science conference.

The material has some very attractive advantages. The handling of glass is known to be very durable when needed. When no longer needed, glass can be melted down and made into new glass. The material is one of only a few that do not notably degrade when melted and reformed. Glass is also easy to transport, even if a little care is required, and it can be formed into any shape desired. Raw materials are also readily available and comparatively cheap.

Microsoft's ideal vision is huge storage banks of transparent data carriers in the cloud, which can remain in operation as long as the data center itself remains intact. To store more data, new disks are simply added. Glass is also insensitive to electromagnetic compatibility (EMC), which means that a digital legacy could finally become truly permanent.

Write Levels on a Glass Plate

Glass data carriers are written by a femtosecond (10^{15}) pulse laser that penetrates the glass during writing. The data is therefore protected by the surface of the glass medium. Microsoft writes up to 200 separate data layers in a square glass medium about the size of a CD.

The laser creates line-like markings in the glass, which Microsoft refers to as voxels. Depending on the orientation of the polarization, the voxels can be rotated, which means a voxel can represent more than one bit. Each level can be described separately with voxels. Thanks to cryptographic procedures, the written data (1) is redundant and (2) cannot be read without decryption. It also cannot be manipulated retrospectively, which would make technologies such as blockchain superfluous, at least in part.

Around 1,000 voxels per level are bundled into a square field that, with all of its levels, is the basic unit when reading the data carrier with a polarization-sensitive microscope in daylight conditions. The different orientation of the voxels polarizes the incident light, and this polarization is measured. When reading, it is not the medium that moves but the reading beam (beam steering) to minimize the risk of breakage.

With this method, reading speeds similar to the latest Linear Tape Open (LTO) standards can be achieved in the laboratory. Microsoft claims to have already read square media completely from corner to corner in the lab. The read signals are decoded for everything on a square 1,000-voxel field on the data carrier by artificial intelligence and machine learning.

Highly Modular Design

Microsoft has also already given some thought to system design. The three important components are the medium shelf, along with autonomous handling robots, and the read and write systems. Unlike today's tape systems, the three work completely independently of each other, which means that a highly modular system can be designed to reflect the requirements of the target use environment. Conceivably, you could operate write-only units in one area and then transport the written media to another area where they are archived and read as required. Write technology then is not necessary, which lends itself to a heterogeneous archive structure. A preliminary study showed that data archives differ greatly in terms of the volume of data written and read, depending on their purpose. In principle, however, the volume of data read decreases with the age of the data, which is one reason why the archive medium must be durable but inexpensive.

With modular systems of this kind, any system can be expanded to meet requirements. The racks comprise smaller racks that can be connected in series and set up parallel to each other. Only written media are stored there and are then ready for data retrieval. Each rack level has a rail along which a handling robot moves. The robot retrieves written media from the racks, takes them to the reading systems, and returns them to storage once the read is complete. The robots can switch from a specific rail on the rack to the one above or below it by a tilting mechanism. In other words, if a robot fails or is blocked, the entire rack is not necessarily down – only the area directly below that robot.

At the 2024 OCP Summit, Microsoft presented its groundbreaking proposal to the open source community. The call went out to developers in all companies affiliated with OCP to develop the required components as quickly as possible and make them ready for use. Microsoft did not provide any forecasts on the time required for this step.

A Protocol for the New Age

Another open source innovation, although (yet) outside the scope of OCP, was recently presented to the general public on an IT press tour in Madrid: the Zenoh data protocol [3]. This network protocol is expected to play a similarly important role in data transport to IP in the development of the Internet as Kafka does today for real-time data streaming, or to certain protocols in robotics. Zenoh could replace important protocols in both worlds.

The originator of the idea is Italian protocol manufacturer ZettaScale. Around a third of the approximately 60 employees there are involved in the further development and implementation of Zenoh. The development work is taking place in France and the Netherlands. Historically, the company is an ADLINK Technology spin-off. The strategic partner is TTTech Auto, an Austrian company that specializes in automotive, robotics, and IoT.

Zenoh is intended to put an end to the patchwork of protocols typical of today's edge-to-core implementations. This protocol confusion is the main reason for delayed IoT and robotics implementations in the industrial sector and for the high energy consumption caused by data transport.

In contrast to other solutions, Zenoh is not host-centric, but data-centric and consistently decentralized. It combines the handling of data at rest, transported data, and data currently used in computing processes in all areas. Its abstractions are location independent. Queries can be distributed across heterogeneous systems. In terms of the publish-subscribe-query (pub-sub-query) structure, it is similar to REST. The work is well advanced. The protocol is due for certification in line with ISO 26262 Automotive Safety Integrity Level D (ASIL D, the highest integrity requirements) in the near future. The standard describes the safety of autonomous vehicles. Zenoh has already been used in the US Indy Autonomous Challenge, a race for autonomous vehicles, as well as in autonomous vehicles and trains run by German national railway operator Deutsche Bahn. The field of application in each case is autonomous communication between vehicles or robots and their environment.

An evaluation by Open Robotics showed that Zenoh performed best of all the protocols. Compared with the Data Distribution Service (DDS) protocol, which is often used in this environment, it drastically reduces the number of discovery packets and therefore increases the number of data packets transported during a unit of time – particularly in multicast environments.

Zenoh offers native libraries for major modern programming languages (e.g., Rust, C or C++, Python, JavaScript, REST, etc.). Shared memory is supported. The protocol is based on the data layer, but can also work with existing network or transport layers. Thanks to its tiny overhead of just 5 bytes, it also works in the smallest embedded and other environments, as well as on Linux, macOS, Windows, and – currently as an alpha version – QNX. Potential platforms include Arduino, ESP32, Mbed, Zephyr, and others, along with automotive micro-synthetic aperture radar (MICROSAR) by AUTOSAR Classic.

The protocol also works on any topology: peer-to-peer, routed, or with a broker. The pub-sub-query mechanisms satisfy a data request from the next node that has the data available, minimizing transport routes for data by doing so. Requests can be answered simultaneously by all nodes on which the required data is available, because they no longer have to pass through a central cloud, but only to the sources of the required data.

If a car has an accident when crossing an intersection and the accident needs to be investigated at a later time, the data from each vehicle (and from traffic lights or other optical sensors located at the scene of the accident) could be provided automatically to the investigation team for the exact time of the accident. The investigation team does not have to know exactly which device they need to access at the scene of the accident, because the black boxes in the vehicles might no longer have any information about the critical moment.

Datasets that can be queried (queryables) can also extend across replications or partitions but are still recognized when queries are answered. Requests to storage are represented by a queryable and a subscriber that orders data from there.

Zenoh's energy efficiency, with its improvements of several orders of magnitude, is the result of greater process efficiency, because only a small amount of energy is required to assemble and process a dispatched packet. Moreover, only the necessary payload bytes are sent, with very little overhead and by the shortest possible route.

ZettaScale now also offers an online platform, similar to Confluent for Kafka, that can be used to provide, manage, and monitor cloud-to-microcontroller infrastructures.

Conclusions

Approaches such as ZettaScale's Zenoh and glass storage media clearly demonstrate that storage and IT have by no means reached the end of their potential for taming the flood of data. Instead, they may well be setting off for completely new shores to handle the tasks that await them. Admins can see some promise in battling today's storage and protocol worlds at data centers, as the approaches currently being pushed toward maturity by OCP show. Be ready to keep your eyes peeled so you can get involved as soon as the threshold of practical maturity has been crossed. ■

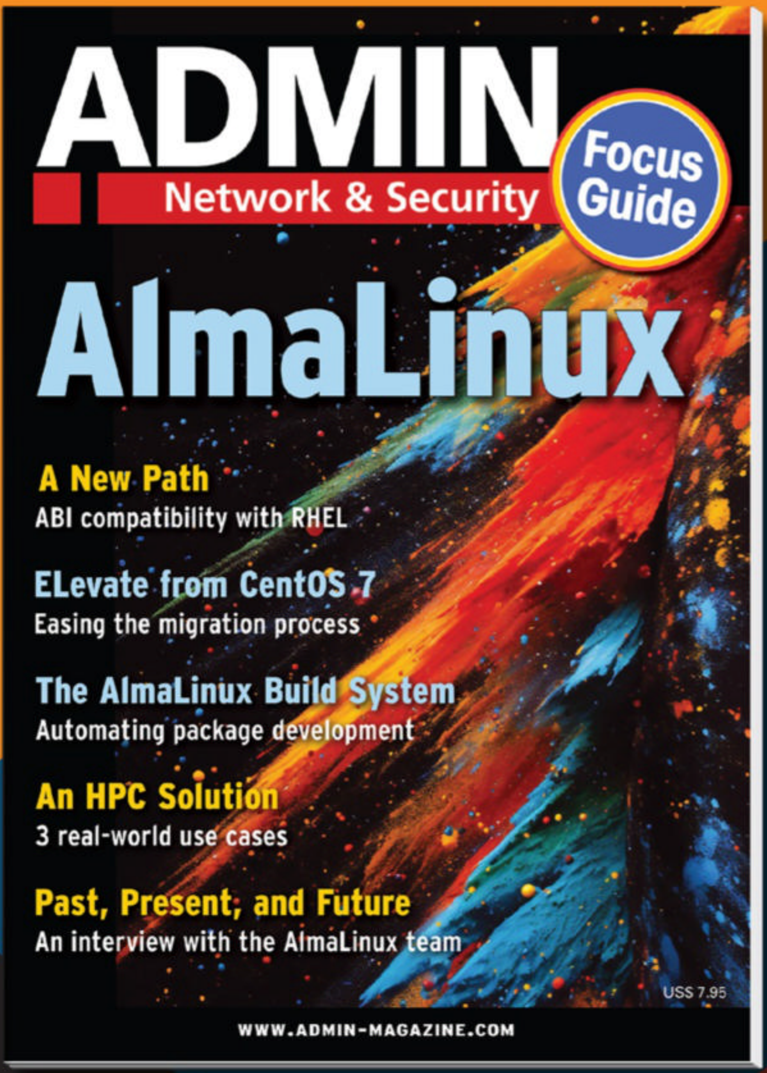
Info

[1] SNIA: flexible data placement: [https://www.snia.org/educational-library/flexible-data-placement-open-source-ecosystem-2023]

[2] Project Silica: [https://www.microsoft.com/en-us/research/project/project-silica/]

[3] Zenoh: [https://zenoh.io]

Inside AlmaLinux



ADMIN

Network & Security

Focus Guide

AlmaLinux

A New Path
ABI compatibility with RHEL

ELevate from CentOS 7
Easing the migration process

The AlmaLinux Build System
Automating package development

An HPC Solution
3 real-world use cases

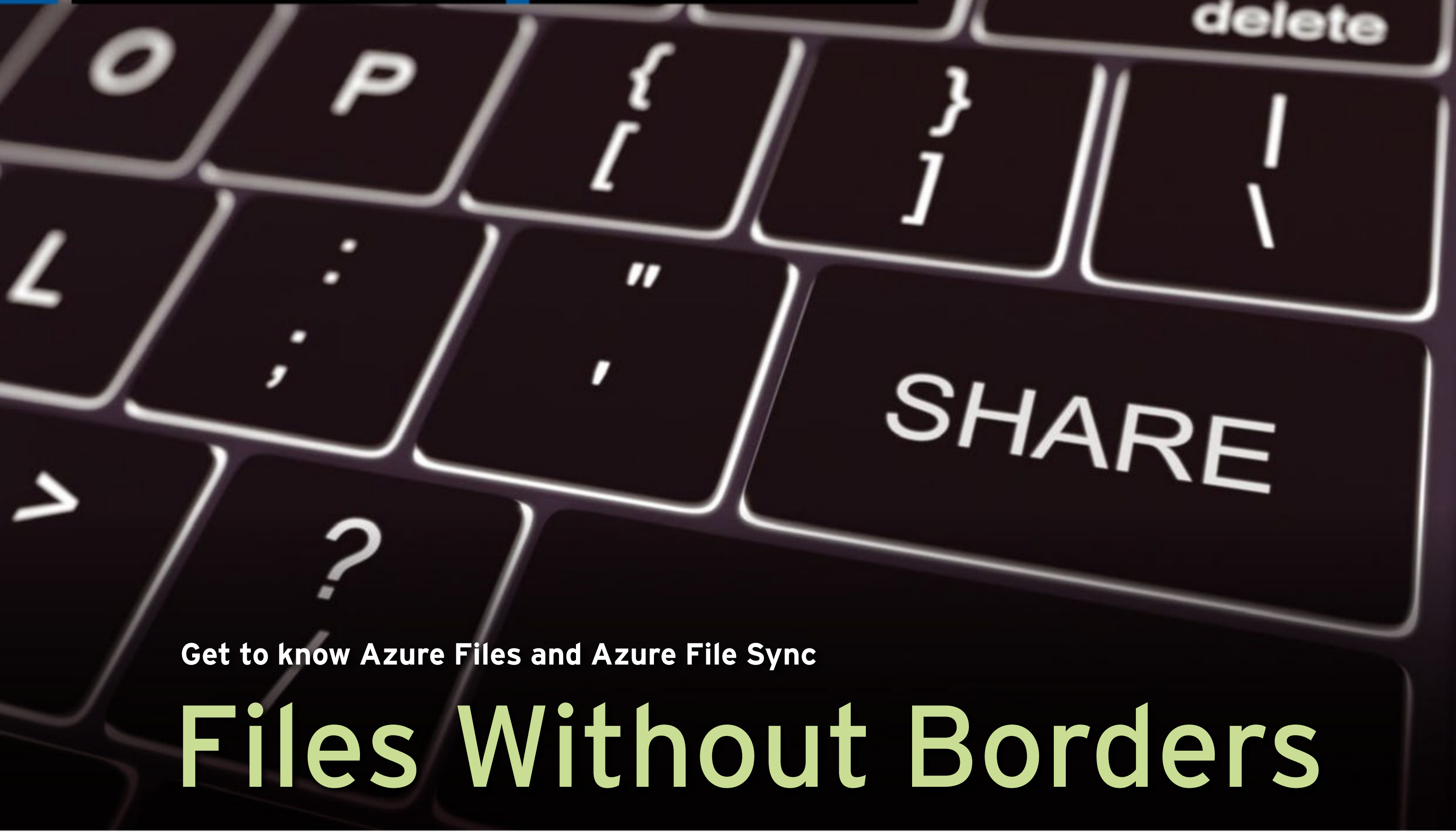
Past, Present, and Future
An interview with the AlmaLinux team

US$ 7.95

WWW.ADMIN-MAGAZINE.COM

Download this free focus guide and get an introduction to AlmaLinux OS, a forever-free, community-based enterprise Linux distribution.

https://bit.ly/AlmaLinux-focus-guide



Get to know Azure Files and Azure File Sync

Files Without Borders

Azure Files and Azure File Sync are Microsoft's classic file shares for clients on Windows, Linux, and macOS in the cloud. We introduce you to their services and guide you through their setup. By Christian Knermann

The unabated trend toward cloud-based infrastructures has given rise to many new web applications and services that differ in terms of their architecture from the legacy client-server model. Storage management is no exception, with object storage for virtually unlimited volumes of data on the rise in the form of services that offer compatibility with Amazon Simple Storage Service (S3) – or at least similar functionality. Microsoft is also looking to become a serious player in this field, offering blob containers, queues, and tables as storage for cloud-native applications in the Azure cloud.

Legacy CIFS, SMB, and NFS

LANs in corporate environments are still inhabited by business-critical applications that adhere to traditional principles and rely on conventional file shares. Windows in particular relies on the SMB/Common Internet File System (CIFS) protocol, but macOS and Linux (with the *cifs-utils* package) also use it. Microsoft coined the original term "Common Internet File System," and it is still a part of everyday life. On closer inspection, however, CIFS only refers to the first standardization proposal that Microsoft submitted to the Internet Engineering Task Force and not the SMB protocol, which has now reached version 3.1.1. Therefore, I will only be referring to SMB in this article. The free Samba program package allows Linux and Unix derivatives to act as SMB servers and even (to a limited extent) as domain controllers in Active Directory (AD), although they rely on NFS by default.

Azure Files is Microsoft's way of implementing serverless, fully managed file shares in the cloud to integrate clients on Windows, macOS, and Linux over the Server Message Block (SMB) or Network File System (NFS) protocol [1]. (See the "Legacy CIFS, SMB, and NFS" box.) Users can still rely on familiar approaches (e.g., the graphical File Explorer, the command line, PowerShell, and Linux Shell). In a best case scenario, clients and end users will not even notice a difference compared with working on a legacy local file server. Moreover, the service comes with a REST API, referred to as FileREST API in the online documentation, to enable access over HTTPS.

Organizations pursuing a cloud strategy cannot simply replace existing SMB- and NFS-based applications and services with state-of-the-art alternatives overnight, which raises the question of how to migrate existing applications that are based on these protocols to the cloud. Although you could simply move servers you previously operated on-premises to virtual machines in the cloud, Microsoft's Azure Files [2] offers a leaner alternative.

Getting Started on the Azure Portal

To get started with Azure Files, you first need a storage account in the Azure cloud. The quickest way to set things up is to select *Create a resource* at the top of the vertical menubar on the left of the Azure portal. You'll find *Storage accounts* in the list of services and the Azure Marketplace where you can select *Create* | *Storage account*. Alternatively, you could use PowerShell or the Azure command-line interface (CLI) to create a storage account with shares, but I will continue to follow the wizard for the time being.

Select your subscription and, if already available, the target resource group. If you are looking to map a use

Lead Image © moovstock, 123RF.com

case fully in the cloud, the resource group containing the SMB or NFS clients is the obvious choice. When you name the storage account, note that this part of the URL is used for access over public networks, which means it must be globally unique. If you try something generic, such as *azurefiles* or *azfiles*, the portal will respond with an error message because the name is already in use. In this example, I use *cloudyfilethings* as the account name for the examples.

Differences in the Azure Accounts

In terms of performance, Microsoft distinguishes between standard (a universal v2 account that relies on hard disk storage) and premium (also known as FileStorage) accounts. The standard account, according to Microsoft, covers most use cases with up to 1,000 input/output operations per second (IOPS). Besides file shares, this type of account offers blob containers, queues, and tables as storage resources and supports locally redundant (LRS), zone-redundant (ZRS), geo-redundant (GRS), and geo-zone-redundant (GZRS) storage [3].

The premium account uses solid-state disks (SSDs) as the data carriers and is recommended for applications that require particularly low latency and more than 1,000 IOPS. A FileStorage account can exclusively contain either file shares, page blobs, or block blobs, but no other storage resources at the same time, and only offers the LRS and ZRS redundancy types.

Another important difference is not immediately apparent. Standard accounts only support versions 2.1 to 3.1.1 of the SMB protocol, whereas NFS is reserved for premium accounts. Also note that Azure Files currently only supports NFS up to version 4.1 and that restrictions apply. For example, functions such as delegations and callbacks of all kinds, Kerberos authentication, and encryption during transfer are missing.

Whereas Azure Files for SMB relies on version 2 of the NT LAN Manager (NTLMv2) protocol to support identity-based authentication by Kerberos and shared keys, only host-based authentication is available for NFS access. Also, Azure Files only supports optional Internet access for SMB from version 3.0 upward and not for NFS. Therefore, I will focus on SMB shares herein.

Regardless of performance, redundancy also influences the maximum size of file shares. Locally redundant and zone-redundant accounts can hold up to 100TB with the optional large file shares, whereas geo- and geo-zone-redundant accounts are restricted to a maximum of 5TB. You can also enable support for large file shares at a later date, but you cannot undo this once the option has been enabled. In terms of costs, you can configure a lower maximum capacity for individual shares at a later date.

On the *Advanced* tab, keep the default settings. TLS 1.2, the highest available minimum version for transport layer security, is already enabled here. Optionally, you can restrict copy operations to storage accounts in the same Microsoft Entra ID Tenant or accounts with private endpoints to the same virtual network. At press time, Microsoft was still tagging this function as a Preview.

Access over Public Networks

Further options for restricting access can be found on the *Networking* tab. Public access from all networks is active by default, and that includes from the Internet. Alternatively, you can restrict public access to selected virtual networks and IP addresses or disable it entirely so that only private access remains, for which you need to configure a private endpoint connection. The preselected Microsoft network routing preference means that data traffic is routed as near as possible to the client in the Microsoft cloud, whereas Internet routing means routing close to the Azure endpoint.

Even when public access is active, it does not eliminate the need for further network configuration tasks, depending on the application, because many company networks, but also Internet providers, block TCP port 445, which is used for SMB. If your SMB clients themselves reside in the Azure cloud, you won't have a problem. For local clients you will need a virtual private network (VPN) or ExpressRoute connection between your local network and the Azure network with a private endpoint for Azure Files within your Azure virtual network. Microsoft explains what you need to think about in terms of network operation in the online documentation [4].

Data Protection

The *Data protection* tab does not address data protection in the legal sense; instead, it means protection against accidental deletion. Microsoft prescribes a retention period of seven days for file shares. It is important to note that this only refers to the deletion of the file share as a whole and not to individual data in the share. In other words, the protection function is no substitute for backups and snapshots, which you will want to configure for each file share.

On the tab for encrypting inactive data, you can choose Microsoft managed keys (MMKs) or customer managed keys (CMKs). In the first case, Microsoft has the keys and takes care of rotating them regularly. You can also manage your own keys, but then you have to take care of the rotation process yourself. You also need to authorize Azure Files to access your keys to enable read and write requests from clients.

Optionally, you can add tags to the storage account to make it easier to categorize resources and display consolidated billing in large environments. The final step is to verify the selected settings and create the storage account; it will appear a short time later under *Storage accounts* in the Azure portal's main menu.

When you select the object, the portal shows you a submenu where you can modify the advanced options selected during setup, along with the network, data protection, and encryption settings. This process only works if the function dependencies allow it. For example, enabling an option for large file shares restricts the choice of redundancies.

Shares with Backups and Snapshots

Select *File shares* under the Data storage section of your storage account and create an initial share; I use *cloudysharethingy* as the name for this example. Besides defining a name, you can edit the tier in the general information section, which is all about defining the primary purpose of the file share. The *Hot* tier is used for universal file shares that support direct interaction with end users working in a team and as a basis for Azure File Sync.

The *Cool* tier optimizes storage for use as an online archive, and the *Transaction optimized* tier is recommended for transaction-intensive server applications without particularly strict latency requirements. *Premium* is only available for storage accounts that use SSD-based premium storage. I selected *Hot* for my first share.

On the *Backup* tab, Microsoft enables daily data backups by default and stores these for 30 days. Alternatively, you can configure an individual backup policy. The available options are easy to understand. Instead of a daily backup, you can opt for backups every four, six, eight, or twelve hours. Once the share has been created, you can then manually initiate an initial backup under the Operations section with the *Backup* tab or simply wait for the first scheduled backup to run. Once a backup is available, you can either restore the entire share or retrieve individual folders and files.

In addition to backups, you can create up to 200 incremental snapshots for each share by selecting *Snapshots* under the Operations section; you can store these files for up to 10 years for read-only access.

Connecting Windows, Linux, and macOS

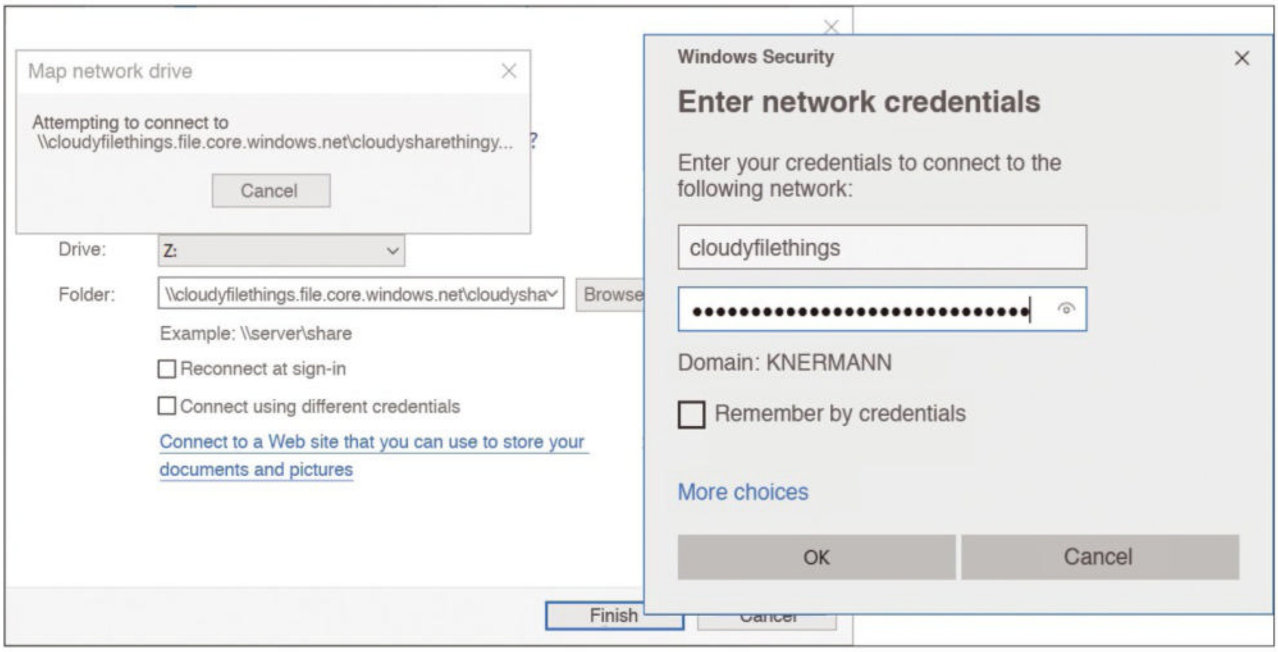
To begin, I'll start by connecting the share manually on a client. If you do not have any other form of identity management (I will return to this in a moment), you will need an access key as a password. This key can be found under *Security + networking* | *Access key* at the storage account level. The Azure portal offers you two keys that you can use interchangeably and rotate independently, which means you can replace one of the keys while you are using the other.

Now connect a network drive on a Windows client in Explorer, for which you will need the fully qualified domain name (FQDN) of the storage account and the share as the target; in this case, it is *\\cloudyfilethings.file.core.windows.net\cloudysharethingy*. Enter the name of the storage account in the field for the username and one of the two access keys as the password (Figure 1). You will now see the share in the cloud as a network drive in Explorer. As an alternative to the graphical dialog, you can pop up a command line on Windows and type

```
net use Z: ↩
  \\cloudyfilethings.file.core.windows.net\↩
  cloudysharethingy ↩
  /USER:cloudyfilethings "<access key>"
```

or you can use PowerShell:

```
New-SmbMapping ↩
  -LocalPath Z: ↩
  -RemotePath "\\cloudyfilethings.file.↩
   core.windows.net\cloudysharethingy" ↩
  -UserName "cloudyfilethings" ↩
  -Password "<access key>"
```



Map network drive

Attempting to connect to \\cloudyfilethings.file.core.windows.net\cloudysharethingy...

Cancel

Drive: Z:

Folder: \\cloudyfilethings.file.core.windows.net\cloudysha

Browse

Example: \\server\share

□ Reconnect at sign-in

□ Connect using different credentials

Connect to a Web site that you can use to store your documents and pictures

Finish Cancel

Windows Security

Enter network credentials

Enter your credentials to connect to the following network:

cloudyfilethings

••••••••••••••••••••••••••••••

Domain: KNERMANN

□ Remember by credentials

More choices

OK Cancel

Figure 1: Accessing Azure Files with File Explorer is hardly any different from accessing local file shares.

Folders and files that you save in the file share on Windows from this point on will also appear when you use the *Browse* item to view the share on the Azure portal. The *Connect* option in the header of the view also supplies scripts for Windows, Linux, and macOS, which you can use to integrate the share in the future without any manual overhead (Figure 2).

Granular Authorizations – Only with IAM

Note that although the connection is quickly configured with an access key, it does not support granular authorization. In other words, when you mount a share, you automatically have full administrative authorization. Microsoft only recommends this type of access in conjunction with VPN connections and private endpoints, but not for public access over the Internet. In the share itself, you cannot easily assign NTFS permissions to users and groups from your local domain because the storage account in the cloud does not recognize these IDs. Therefore, the use of an access key to integrate a share is only suitable for a few server applications that do not require direct interaction with end users; it is definitely not going to be your option for teamwork. Fortunately, Microsoft offers several options for granular identity and authorization management, including integration with Entra ID, the cloud-based identity and access management (IAM) tool formerly known as Azure AD, and integration with a local AD [5], which requires synchronization of the local AD with the cloud in Microsoft Entra Connect, the product formerly known as Azure AD Connect.

Setting Up Synchronization Services

SMB shares that you migrate to the cloud with Azure Files look just like local resources to end users, but what do you do if the bandwidth is too low or the latency too high, and users complain about poor performance when they try to work with data stored there? This is where Azure File Sync enters the scene. It uses one or more Windows servers – that can even reside at different sites – for local caching. This arrangement cleverly combines the advantages of centralized storage in the cloud with the faster speed of local access.

On the Azure portal, select *Create a resource* from the main menu, search for *Azure File Sync*, and set up a new resource of this type. The first step is to determine in which subscription and resource group the service will be located and give it a name (*cloudysyncthingy* in this example). As with the storage account, you can allow access by all networks or only by private endpoints; proceed to tag the resource, if so desired, and create it. Now switch to the new resource. In the submenu, select *Sync group* and create a new group. Besides a name, you also need to select the subscription, the target storage account, and a share that will play a bidirectional role as the source and target for synchronization with your local file server.

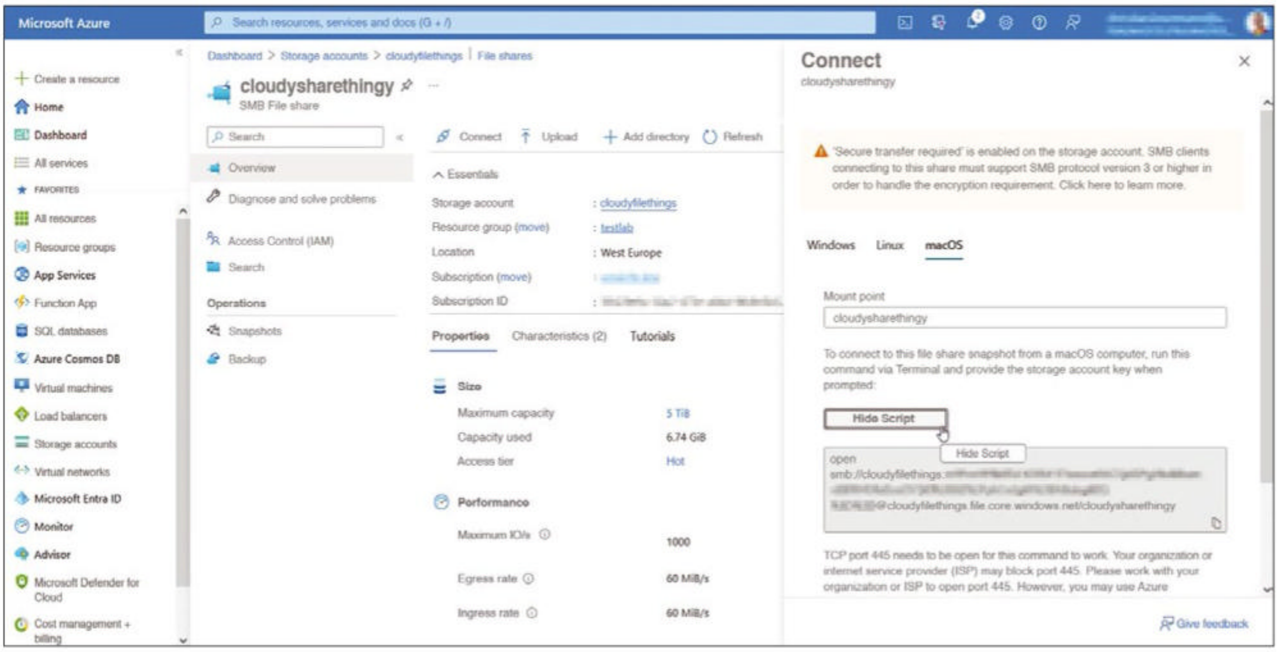
Configuration for Windows Server

If not already in place, install the PowerShell module for Azure Resource Manager on the local file server by running

```
Install-Module -Name AzureRM
```

in an admin PowerShell, which prompts you for the NuGet provider and permission to use the *PSGallery* repository, both of which you confirm.

Now download and install the Azure File Sync agent, which is available



Storage account : cloudyfilethings
Resource group (move) : testlab
Location : West Europe
Subscription (move) : [redacted]
Subscription ID : [redacted]

Maximum capacity 5 TiB
Capacity used 6.74 GiB
Access tier Hot

Maximum IO/s 1000
Egress rate 60 MiB/s
Ingress rate 60 MiB/s

'Secure transfer required' is enabled on the storage account. SMB clients connecting to this share must support SMB protocol version 3 or higher in order to handle the encryption requirement. Click here to learn more.

Mount point: cloudysharethingy

To connect to this file share snapshot from a macOS computer, run this command via Terminal and provide the storage account key when prompted:

```
open
smb://cloudyfilethings:[redacted]@cloudyfilethings.file.core.windows.net/cloudysharethingy
```

TCP port 445 needs to be open for this command to work. Your organization or internet service provider (ISP) may block port 445. Please work with your organization or ISP to open port 445. However, you may use Azure

Figure 2: The Azure portal offers scripts to help users mount shares on Windows, Linux, and macOS.

for Windows Server 2012 R2, 2016, 2019, and 2022 [6]. The setup routine shows the ubiquitous license agreement, prompts you for the installation path, and optionally uses a proxy server and Microsoft Update.

The configuration wizard connects to either the Azure cloud or to separate cloud instances for the Chinese market or the US government. As a rule, *AzureCloud* is going to be the right choice. After logging in to the cloud with an admin account, select your subscription, the resource group, and the storage synchronization service. The wizard then registers your server in the cloud and tests the network connection. If both actions are successful (Figure 3), the server appears on the Azure portal in the storage synchronization service resource in the *Registered servers* option under the Sync section.

Next, navigate to the view of the previously created synchronization group, and under Server endpoints, select *Add server endpoint*. In the next step, set up your local server and a path on the server as the remote synchronization point.

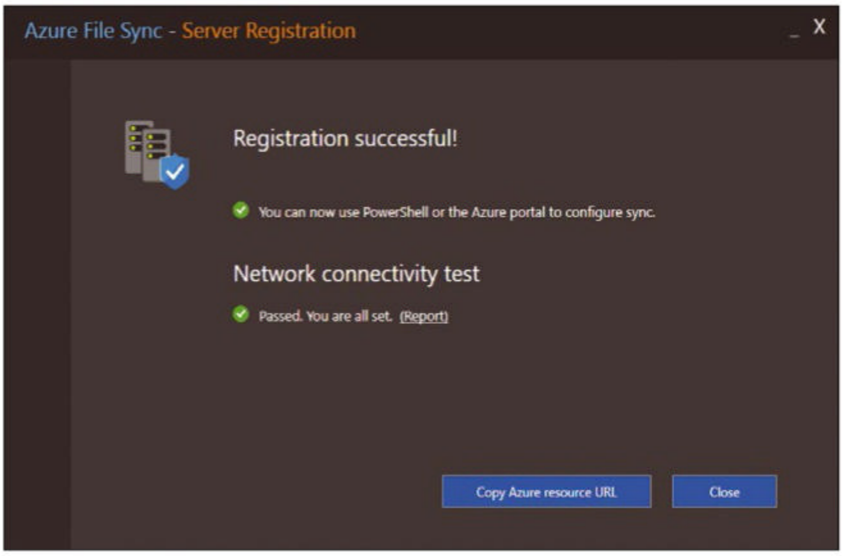
If you want to transfer large volumes of data, Microsoft also supports the Data Box and Data Box Heavy methods [7] as alternatives to network-based synchronization. In this case, the data then finds its way to the cloud by means of Microsoft sending you a special Data Box storage device to which you copy the desired data offline before mailing the device back to Microsoft.

With cloud tiering disabled by default, the service synchronizes all data between your local server and the cloud. If you enable cloud tiering, the service only stores hot data locally, whereas cold data remains in the cloud. If data is already available both locally and in the cloud before the initial sync, Azure File Sync keeps both files in the event of conflicts. Alternatively, you can define the local server as the lead, which means that files that already exist in the cloud are overwritten in the event of file conflicts.

Conclusions

Despite trends toward the cloud, many use cases still rely on legacy SMB and NFS shares on the local network. Azure Files sees Microsoft support file shares that do not require conventional file servers in the cloud. From the perspective of clients on Windows, Linux, and macOS, users are unlikely to notice any difference.

However, you will need to integrate your local AD with the Azure cloud to implement end-to-end management for identities and authorizations. Where bandwidth or latency requirements dictate, additionally implementing the Azure File Sync service ensures continuous synchronization of local file servers and file shares in Azure Files. ■


Azure File Sync - Server Registration

Registration successful!

✓ You can now use PowerShell or the Azure portal to configure sync.

Network connectivity test

✓ Passed. You are all set. (Report)

Figure 3: The Azure File Sync agent bridges the gap between local file servers and shares in the cloud.

Info

[1] Introduction to Azure Files: [https://learn.microsoft.com/en-us/azure/storage/files/storage-files-introduction]

[2] Azure Files: [https://azure.microsoft.com/en-us/products/storage/files]

[3] Planning for redundancy: [https://learn.microsoft.com/en-us/azure/storage/files/storage-files-planning#redundancy]

[4] Network operation considerations: [https://learn.microsoft.com/en-us/azure/storage/files/storage-files-networking-overview]

[5] Managing identities: [https://learn.microsoft.com/en-us/azure/storage/files/storage-files-planning#identity]

[6] Azure File Sync agent: [https://www.microsoft.com/en-us/download/details.aspx?id=57159]

[7] Azure Data Box: [https://learn.microsoft.com/en-us/azure/databox/]

Author

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

SFSCON
FREE SOFTWARE CONFERENCE

NOI
TECHPARK SÜDTIROL / ALTO ADIGE

NOV
8TH
9TH

2024

BE PART OF IT!

2 INSPIRING DAYS
1 FESTIVAL
100 SPEAKERS

WWW.SFSCON.IT

NOI TECHPARK
SÜDTIROL / ALTO ADIGE

GRUPPO FOS soluzioni ad alta tecnologia · TELMEKOM NETWORKS · SYMPHONIE PRIME · VATES Open infrastructure made simple · MADE IN CIMA · Zirkonzahn Human Zirconium Technology

1006.org · {catchsolve} · Christian Gapp · endian · ecosteer. · OBUS Information & Technology Group · peer · (studio hug)

NOI TECHPARK
SÜDTIROL / ALTO ADIGE

A.-VOLTA-STRASSE 13A, BOZEN
VIA A. VOLTA 13A, BOLZANO

VISIT US ON
WWW.NOI.BZ.IT

A feature-rich drop-in-replacement for Microsoft Exchange

Exchanging Exchange

Grommunio is a completely open source and fully compatible drop-in replacement for Microsoft Exchange that uncouples your company from Microsoft's cloud strategy and its severe security and data protection issues. By Markus Feilner

Microsoft's Exchange Team published a blog post on August 8, 2024, about decommissioning Exchange Server 2016. Although the intention was definitely not to make people switch to an open source groupware, the announcement was enlightening:

> *Exchange 2016 is approaching the end of extended support and will be out of support on October 14th, 2025. If you are using Exchange Server 2019, you will be able to in-place upgrade to the next version, Exchange Server Subscription Edition (SE), so Exchange Server 2016 will need to be decommissioned at some point. If you plan to stay on-premises, we recommend moving to Exchange 2019 as soon as possible.* [1]

A long list of todos and caveats follows, including reboots after upgrade. No wonder Microsoft's general strategy toward its cloud (Microsoft 365 and Azure) is worrying users, including many groupware administrators who are currently looking for other options. Not so long ago, Australian vendor Atlassian took a similar approach, attempting to force their customers – users of Confluence, Trello, and Jira – into the vendor-owned cloud. Like Atlassian customers, now more and more Exchange administrators are checking the market, and vendors of open source alternatives are reporting growing numbers of customers. However, what's equally stirring up the groupware market is security and data protection. Over the past few years, several severe issues popped up in or around Microsoft Exchange or its Azure Cloud. After Chinese hackers stole a master key to Microsoft's Azure cloud, the US government did not spare harsh words, calling out Microsoft for "shoddy security, insincerity" and "cascades of security failures" [2].

MAPI: The Curse of Exchange Protocols

Under the pressure of new European Union regulations such as the General Data Protection Regulation (GDPR) or the Cyber Resilience Act (CRA), it's no wonder customers are browsing alternatives. As long as you're only dealing with Linux and open source clients, you will be perfectly happy to use standard open protocols – from IMAP to POP (mail) and from CalDAV (calendars) to CardDAV (contacts). Your client of choice will be Mozilla Thunderbird or KDE's Kontact, whose protocols will work fine with mobile devices, too. However, if your users need or want Windows clients with Microsoft Outlook or a bit more sophisticated support for smartphones, then the situation is different. Outlook and Exchange communicate over a large set of APIs and protocols usually known as the Exchange protocol stack, and in most companies, Outlook is deeply integrated into a number of special applications needed for daily work. Only a few other groupware products can achieve similar features, open source or otherwise.

Vendors usually try to tackle that issue with Outlook plugins, changes to the Windows registry file, or both, but adminstrators tend to dislike any changes to the client's configuration, and both plugins and registry changes have caused problems in the past.

Photo by cdd20 on Unsplash

grommunio: Native Support for Outlook

Grommunio [3] is a startup from Austria that promises a "drop-in replacement" for Microsoft Exchange. No Outlook plugin, no registry entry, no changes to clients, and no need to sit at and click on a client's desktop during migration – so the Vienna company claims. The goal is for every Windows system and almost every smartphone (including Apple clients) to connect seamlessly through standard tools available on every modern system and talk exactly the way they would to Exchange servers. Grommunio's customers confirm that grommunio performs as advertised; Deutsche Telekom is even selling grommunio as part of its Open Source Collaboration Open Telekom Cloud [4].

A Full Stack Re-Engineered

The grommunio developers chose a completely different approach: Instead of plugins and workarounds, they have been (re-)implementing more than 50 protocols and APIs of the Exchange stack in open source and have published all of their work on GitHub. Thus, a grommunio server speaks exactly the Exchange language its Outlook clients expect – natively, but in open source. (See the "From the FSFE and EU" box.)

Because grommunio behaves as an Exchange server would, you simply have no need for a risky plugin or quirky registry entries. Not only is grommunio open source, it also builds on a huge set of trustworthy, renowned, and proven equally stable and secure open source software in the back end.

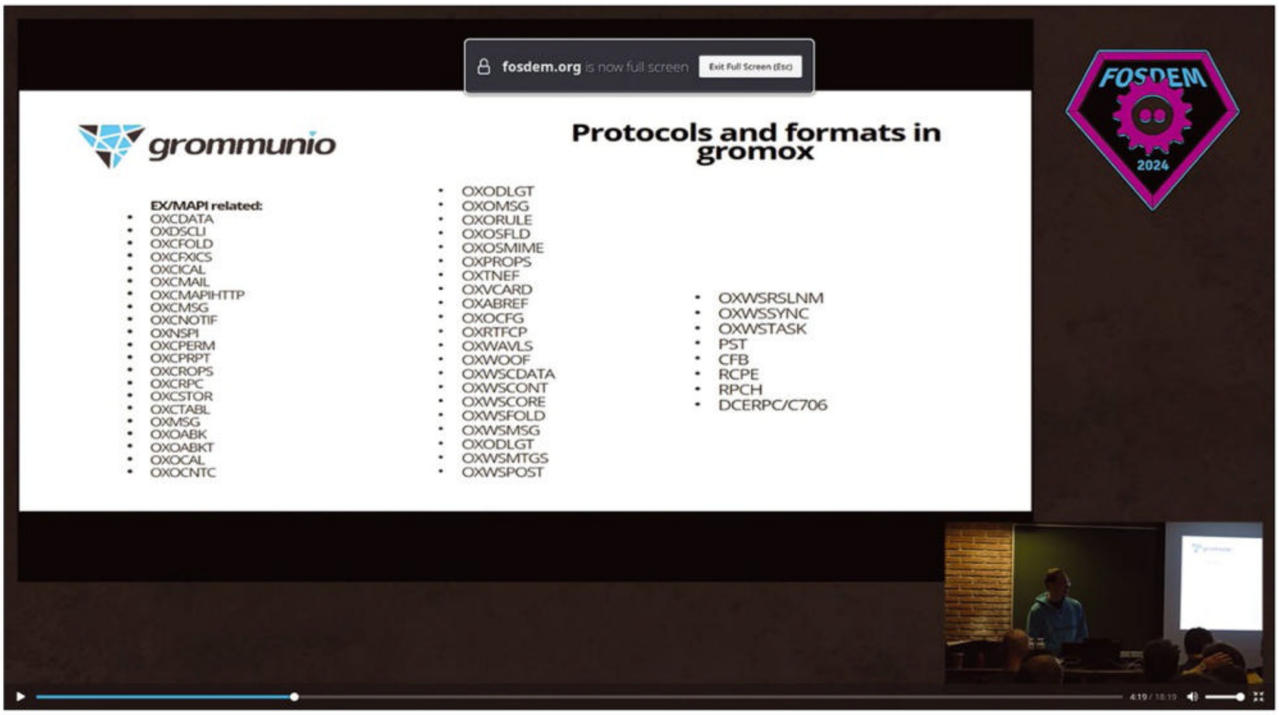
In tedious work spanning more than four years, reading thousands of pages of specifications from Microsoft and testing and occasionally correcting and amending them (Microsoft was always helpful and cooperative), the grommunio developers managed to create an open source server application (gromox) within grommunio that speaks Exchange protocols natively, with a wide range of clients.

Accomplishing that feat took not only a lot of time but also patience, as grommunio's lead developer Jan Engelhardt demonstrated in his presentation at FOSDEM'24 in Brussels (Figure 1), where he went into many, sometimes funny, details of his work [6].

Many Windows and Exchange administrators know the Messaging Application Programming Interface (MAPI), but that term is "somewhat ambiguous," explains Engelhardt. "It is used for concepts as well as for the data mode, programming interfaces, and network protocols on the wire." The grommunio website continues:

From the FSFE and EU

Re-implementing Microsoft standards is only possible thanks to the Free Software Foundation Europe (FSFE) and European Commission: From 2008 to 2012 the FSFE fought through a series of court cases (many antitrust) against Microsoft. The European Commission followed the FSFE's claims: Microsoft must not limit access to its APIs and must release all interoperability information without restrictions. Everything else would be abuse of its monopoly and a violation of the free market's rules. When Microsoft refused to comply, the EU commission cast billions of dollars of fines on them, finally breaking their resistance. Microsoft released the specifications, and it's only for that reason that a project like grommunio is possible. The FSFE has maintained a long record of the case they finally won [5].



Protocols and formats in gromox

EX/MAPI related:

- OXCDATA
- OXDSCLI
- OXCFOLD
- OXCFXICS
- OXCICAL
- OXCMAIL
- OXCMAPIHTTP
- OXCMSG
- OXCNOTIF
- OXNSPI
- OXCPERM
- OXCPRPT
- OXCROPS
- OXCRPC
- OXCSTOR
- OXCTABL
- OXMSG
- OXOABK
- OXOABKT
- OXOCAL
- OXOCNTC
- OXODLGT
- OXOMSG
- OXORULE
- OXOSFLD
- OXOSMIME
- OXPROPS
- OXTNEF
- OXVCARD
- OXABREF
- OXOCFG
- OXRTFCP
- OXWAVLS
- OXWOOF
- OXWSCDATA
- OXWSCONT
- OXWSCORE
- OXWSFOLD
- OXWSMSG
- OXODLGT
- OXWSMTGS
- OXWSPOST
- OXWSRSLNM
- OXWSSYNC
- OXWSTASK
- PST
- CFB
- RCPE
- RPCH
- DCERPC/C706

Figure 1: **Grommunio knows a steadily growing number of Exchange protocols. This list is from a presentation at FOSDEM'24 by Jan Engelhardt [6].**

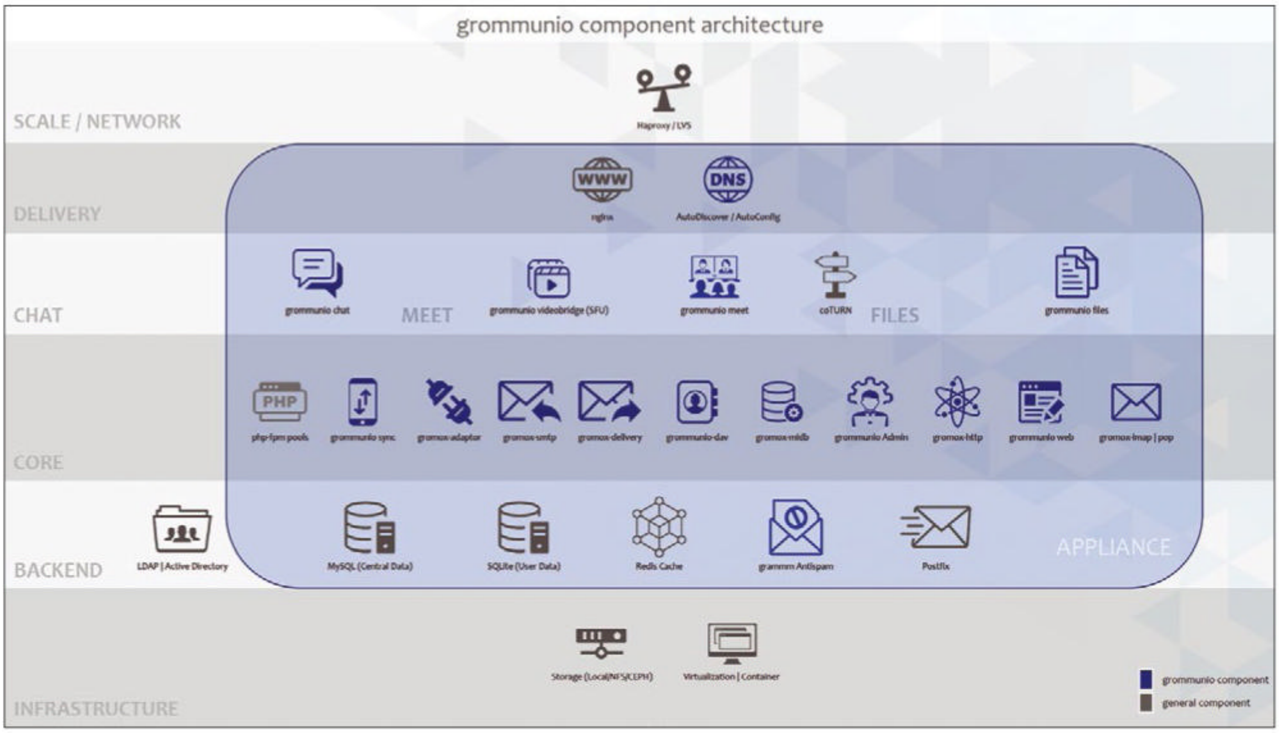
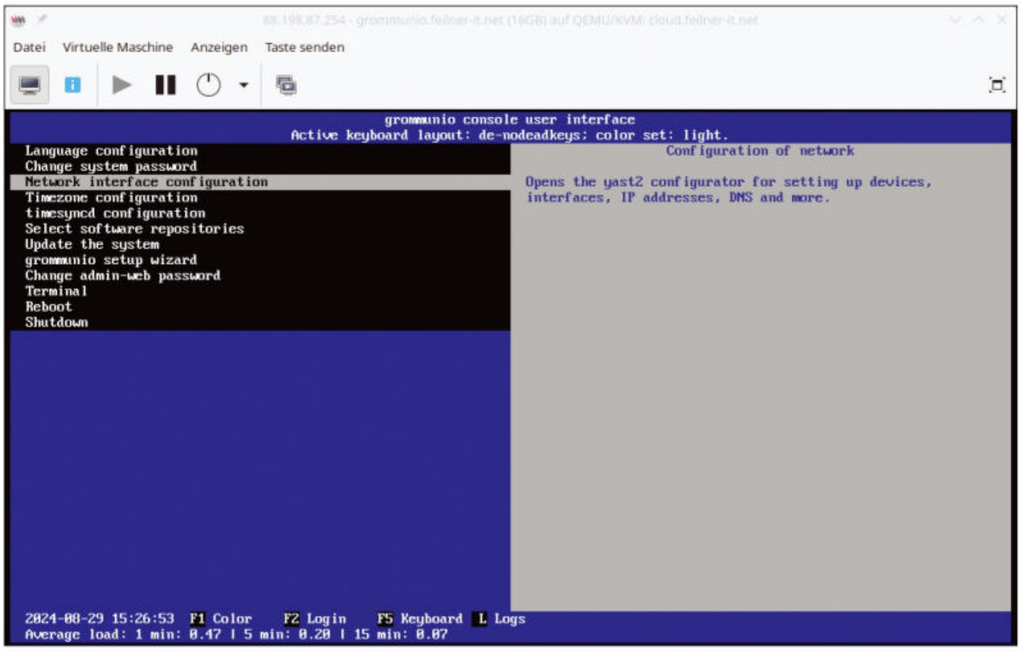


Figure 2: Grommunio is more than just groupware and integrates many other open source tools, from Nextcloud to Jitsi and Mattermost. It also offers archiving, chat, video conferencing, and mobile device management. © Jan Engelhardt, FOSDEM'24 [6]

> *Thanks to Microsoft, all documentation is freely available, the specifications amount to "132 + documents on 8400 + pages, in addition to the Internet mail protocols (i.e. RFC 5322, 5545, etc.) that must be supported anyway."*

Engelhardt and his team at grommunio dived deep into these documents and also helped to fix some problems. This is how grommunio became a contributor to the open stack of specifications used by Microsoft and all its customers. [7] MAPI and Exchange isn't the only language grommunio speaks (Figure 2). Autodiscovery is used for client configuration (just enter your email address and DNS and the server will do the rest for you); Exchange Active Sync (AES) syncs and controls your mobile phones; and Exchange Web Services (EWS) in the world's first open source implementation that connects not only Apple clients but also Linux programs such as Evolution, Thunderbird, and KDE Kontact with the grommunio server [8]. Also, grommunio Meet (a Jitsi implementation), grommunio Chat (Mattermost), grommunio Files (Nextcloud), Archive, Antispam, and directory management (LDAP, Samba) are all integrated and based on standard tools like Postfix and NGINX.

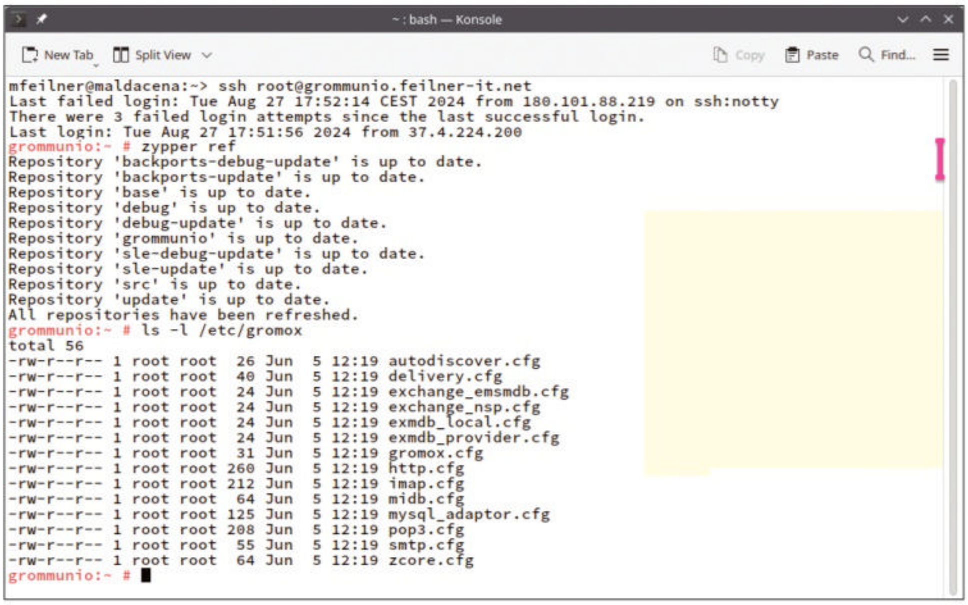

88.198.87.254 - grommunio.feilner-it.net (16GB) auf QEMU/KVM: cloud.feilner-it.net
Datei
Virtuelle Maschine
Anzeigen
Taste senden

```
                    grommunio console user interface
          Active keyboard layout: de-nodeadkeys; color set: light.
 Language configuration                         Configuration of network
 Change system password
 Network interface configuration          Opens the yast2 configurator for setting up devices,
 Timezone configuration                   interfaces, IP addresses, DNS and more.
 timesyncd configuration
 Select software repositories
 Update the system
 grommunio setup wizard
 Change admin-web password
 Terminal
 Reboot
 Shutdown

 2024-08-29 15:26:53  F1 Color   F2 Login   F5 Keyboard  L Logs
 Average load: 1 min: 0.47 | 5 min: 0.20 | 15 min: 0.07
```

Figure 3: The grommunio console interface isn't only used during installation; it offers a fast and easy-to-use access point for basic configuration.

Demo, Docker, and Installation

A grommunio demo on its website, Linux packages for many distributions, and ISO images for download and installation can help you get started. Docker images are available, as well, and all sources can be found on GitHub. SUSE already has RPMs in their repositories (`zypper se gromox`). The ISO images from the company's website are based on openSUSE Leap and can be installed in any virtualization.



```
mfeilner@maldacena:~> ssh root@grommunio.feilner-it.net
Last failed login: Tue Aug 27 17:52:14 CEST 2024 from 180.101.88.219 on ssh:notty
There were 3 failed login attempts since the last successful login.
Last login: Tue Aug 27 17:51:56 2024 from 37.4.224.200
grommunio:~ # zypper ref
Repository 'backports-debug-update' is up to date.
Repository 'backports-update' is up to date.
Repository 'base' is up to date.
Repository 'debug' is up to date.
Repository 'debug-update' is up to date.
Repository 'grommunio' is up to date.
Repository 'sle-debug-update' is up to date.
Repository 'sle-update' is up to date.
Repository 'src' is up to date.
Repository 'update' is up to date.
All repositories have been refreshed.
grommunio:~ # ls -l /etc/gromox
total 56
-rw-r--r-- 1 root root  26 Jun  5 12:19 autodiscover.cfg
-rw-r--r-- 1 root root  40 Jun  5 12:19 delivery.cfg
-rw-r--r-- 1 root root  24 Jun  5 12:19 exchange_emsmdb.cfg
-rw-r--r-- 1 root root  24 Jun  5 12:19 exchange_nsp.cfg
-rw-r--r-- 1 root root  24 Jun  5 12:19 exmdb_local.cfg
-rw-r--r-- 1 root root  24 Jun  5 12:19 exmdb_provider.cfg
-rw-r--r-- 1 root root  31 Jun  5 12:19 gromox.cfg
-rw-r--r-- 1 root root 260 Jun  5 12:19 http.cfg
-rw-r--r-- 1 root root 212 Jun  5 12:19 imap.cfg
-rw-r--r-- 1 root root  64 Jun  5 12:19 midb.cfg
-rw-r--r-- 1 root root 125 Jun  5 12:19 mysql_adaptor.cfg
-rw-r--r-- 1 root root 208 Jun  5 12:19 pop3.cfg
-rw-r--r-- 1 root root  55 Jun  5 12:19 smtp.cfg
-rw-r--r-- 1 root root  64 Jun  5 12:19 zcore.cfg
grommunio:~ # █
```

Figure 4: The ISO image of grommunio comes with a full-featured SUSE Linux underneath. All the well-known tools such as YaST and Zypper are available. Grommunio's configuration can be found and changed in plain-text files.

Figure 3 shows a fresh grommunio installation on a KVM Libvirt system in virt-manager, with the help of the grommunio console user interface. The tool that guides you through the installation is also available after a login; on console 1, press F2 to see the menu-based configuration utility.

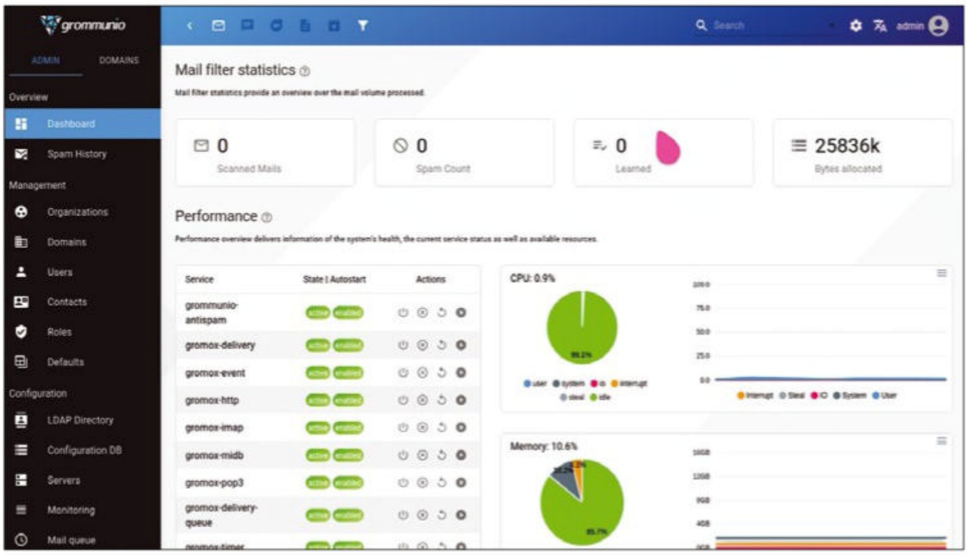
If you're running the grommunio example from the ISO files, you will have the full power of SUSE Linux underneath, including network setup with YaST or the like. No matter which distribution you are using, the grommunio configuration files will always reside in `/etc/gromox` (Figure 4).

Administration

Administering grommunio can be done through Files, the console user interface (CUI), the API, or the web interface (Figure 5). The Admin API for PowerShell (AAPIPS) gives you a grommunio PowerShell interface. Keycloak supplies single sign-on in Active Directory environments, as well (with Simple and Protected Generic Security Service Application Program Interface (GSSAPI) Negotiation (SPNEGO) support). Grommunio Web features OpenID Connect, including support for two-factor authentication (2FA) and Web Content Accessibility Guideline (WCAG) 2.1 compliance.

Grommunio is also the world's first groupware with a multitenant-enabled LDAP server. Administrators of extensive hosted environments, especially, will appreciate that they can configure complex setups with a few mouse clicks: All configuration for the multitenancy needed in large-scale setups can be done in grommunio's admin user interface.

By specifying a separate LDAP back end for their organizations, administrators can establish independent yet seamless integration with individual user's back ends, a configuration especially needed in hosted environments. This immediate, organization-level integration allows hosters and customers to maintain enterprise policies without the otherwise common sync or replica issues and removes many inconvenient steps in authentication and user management.



Mail filter statistics

Mail filter statistics provide an overview over the mail volume processed.

0 Scanned Mails

0 Spam Count

0 Learned

25836k Bytes allocated

Performance

Performance overview delivers information of the system's health, the current service status as well as available resources.

Service	State | Autostart	Actions
grommunio-antispam	active enabled	
gromox-delivery	active enabled	
gromox-event	active enabled	
gromox-http	active enabled	
gromox-imap	active enabled	
gromox-midb	active enabled	
gromox-pop3	active enabled	
gromox-delivery-queue	active enabled	
gromox-timer	[illegible]	

CPU: 0.9%

Memory: 10.6%

Figure 5: Grommunio offers a modern, intuitive, and comprehensive management interface that allows settings for a full domain, services, users, and roles – here in the web version.

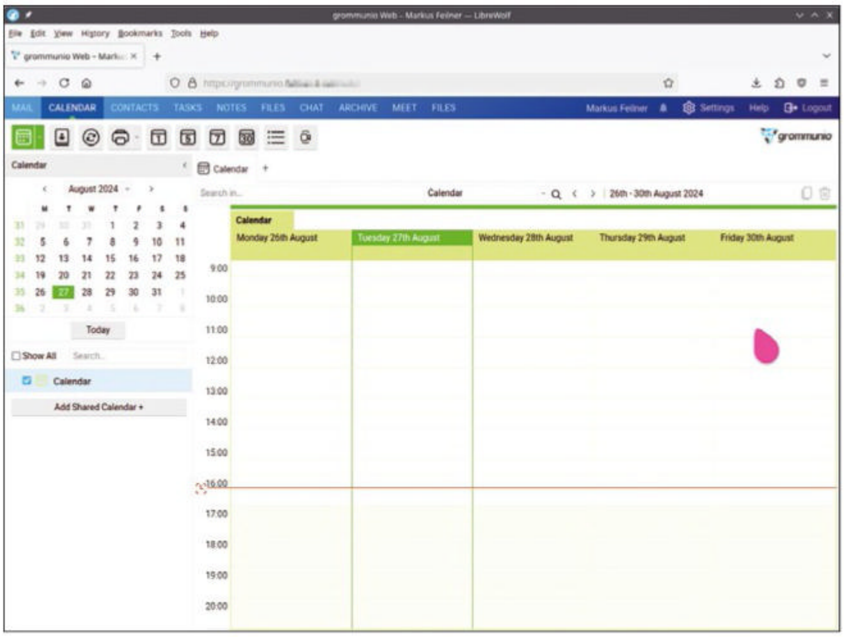


Figure 6: The user web client features mail, calendar, contacts, tasks, notes, file sync, chat, archive, web conference, and more.

Clients and Features

For many users, Outlook on their smartphone will be the number one client platform on which they interact with grommunio, and they won't notice any differences. A user also can use both the web application (Figure 6) and the grommunio desktop application (Figure 7) created with the help of the Electron framework.

The cross-platform Electron makes the desktop app look exactly the same on any platform, creating a consistent user experience and a safe environment independent of a browser. The only difference is that the desktop app runs as a standalone application and has its own menubar where you can select the grommunio server to which you want to connect – making the desktop app a multi-account client. The client is available for Windows and Apple clients, and packages exist for a variety of Linux systems (Debian, RPM, AppImage, and `tar.gz`).

grommunio Next Brings Microsoft Graph

The grommunio developers still aren't done. Their roadmap is ambitious, with plans for the implementation of artificial intelligence (AI) and Microsoft Graph, for which the team is already delivering tech previews.

Grommunio Next [9] is the first open source Graph API web application to provide all the familiar productivity features you need (Figure 8). It is the future main web application for access to your mail, calendar, contacts, tasks, notes, and more.

For those of you who haven't heard of Microsoft Graph yet, it's a unified API platform [10] that connects various Microsoft 365 services, providing a seamless way to access and manage data across tools such as Outlook, Teams, SharePoint, Office (Microsoft 365), OneDrive, and more. With Microsoft Graph, developers can build applications that interact with user data and organizational resources, such as email, files, calendars, and user profiles.

The Graph API offers a single endpoint which streamlines access to information from the entire Microsoft 365 ecosystem, removing the need to work with multiple separate APIs. Graph is kind of a meta-API that allows you to address single fields in an Excel document for read and write with a simple API command. Naturally, such a behemoth of an API is very complex and takes a great deal of time to design and develop. The importance of this task, says Microsoft, lies in Graph's ability to integrate a wide range of data and services, offering organizations a holistic view of their operations.

Real-Time Data

By allowing access to data in real time, Microsoft Graph enables applications to automate workflows, enhance collaboration, and improve productivity. For example, an application can pull calendar information to schedule meetings, manage cloud storage by interacting with OneDrive, or monitor organizational insights. It supports enterprise needs for security and compliance, because developers can access data while maintaining the same governance and control policies

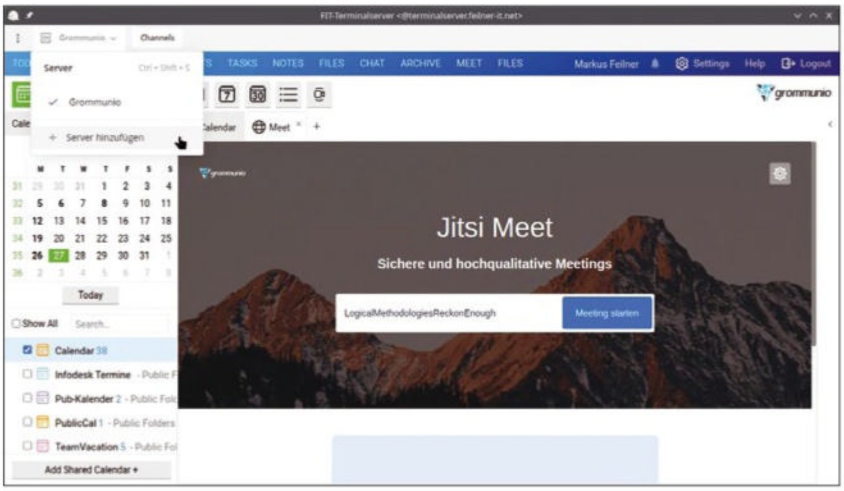


Figure 7: For those who don't want to use Outlook or a browser window, grommunio offers its own Electron-based client that can connect to multiple servers.

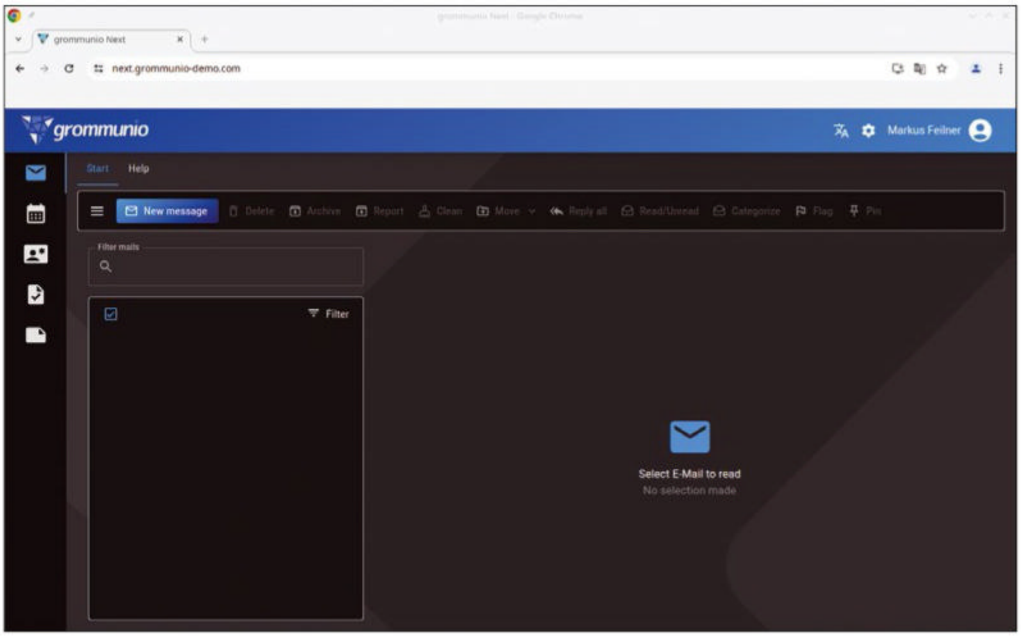


Figure 8: Grommunio is the first open source project to implement Microsoft's future API, Microsoft Graph. On the roadmap, and already available as a Tech preview, you can see the Graph-enabled groupware client connected to grommunio Next.

applied within Microsoft 365 and can help enable AI and automation.

Many analysts see Graph as the future back end for all Microsoft applications, so it's no surprise that many companies are working to integrate it into their products. Grommunio is the first groupware vendor to do so. A tech preview of grommunio Next [11] is already available for testing, and it shows a webmail and groupware client for the browser that is connected to the grommunio server by Microsoft Graph. Again, grommunio's implementation of Graph is the first open source implementation for groupware.

AI and More

Another big topic the grommunio developers are addressing is AI, but on a reasonable scale. The integrated AI tools rather focus on helping the user than on creating hallucinated content. Grommunio Antispam 3.9.0 (released in July 2024) includes GPT-based spam detection, and upcoming are AI-based assistants for email summary generation (available 2024Q4) and for functions such as autotasking in grommunio Web (2025Q1).

Summary

Grommunio offers a fresh new start for open source groupware, and it serves as a real alternative for Microsoft Exchange. With its variety of certifications and compliance with a long list of regulations, customers with strong security and data protection needs will especially benefit from the switch. The version released in August brought certification from the German Federal Office for Information Security (Bundesamt für Sicherheit in der Informationstechnologie (BSI), which is somewhat similar to the US National Institute of Standards and Technology (NIST)).

Accessibility, according to WCAG 2.1 has been on board for some time, and there's more in the pipeline. According to the website, grommunio scales: The vendor claims to be faster than Exchange and more efficient than any other open source groupware. Seems like it's time for a test run. ■

Info

[1] Decommissioning Exchange Server 2016, Microsoft, August 2024: [https://techcommunity.microsoft.com/t5/exchange-team-blog/decommissioning-exchange-server-2016/ba-p/4214475]

[2] Scathing federal report. NBC News, The Associated Press, April 2024: [https://www.nbcnews.com/tech/security/scathing-federal-report-rips-microsoft-shoddy-security-insincerity-res-rcna146177]

[3] grommunio: [https://www.grommunio.com]

[4] Telekom Open Source Collaboration: [https://grommunio.com/open-source-collaboration/]

[5] FSFE and EU vs. Microsoft: [https://fsfe.org/activities/ms-vs-eu/]

[6] Engelhardt, J. "Exchanging Microsoft: Implementing 27 MS Exchange Protocols & APIs in OSS with grommunio." Fosdem'24, Brussels, Belgium, February 3-4, 2024: [https://fosdem.org/2024/schedule/event/fosdem-2024-2731--servers-exchanging-microsoft-implementing-27-ms-exchange-protocols-apis-in-oss-with-grommunio/]

[7] grommunio @FOSDEM 2024: [https://grommunio.com/jan-at-fosdem-2024/]

[8] Exchange Web Services and Linux clients: [https://grommunio.com/grommunios-exchange-web-services/]

[9] grommunio Next demo: [https://next.grommunio-demo.com/]

[10] Microsoft Graph overview: [https://learn.microsoft.com/en-us/graph/overview]

[11] grommunio Next tech preview: [https://github.com/grommunio/grommunio-next]

Author

Markus Feilner is a consultant for open source strategies in Regensburg, Germany, with experience working with Linux since 1994. Markus is now grommunio's Open Source Ambassador and previously was deputy editor-in-chief for the German-language *Linux-Magazin*. His company, Feilner IT, focuses on solving problems on OSI Layers 8 to 11.

Improved logging in Samba Winbind

Keeping Track

In Winbind v4.17, the Samba team has addressed the complexity of and difficulty in troubleshooting the logging service that allows Linux systems to join an Active Directory domain. By Thorsten Scherf

The Winbind service offers various services for the name service switch (NSS) and pluggable authentication modules (PAMs). On the Windows side, Winbind communicates with the Local Security Authority (LSA), Netlogon, and Lightweight Directory Access Protocol (LDAP) services of a domain controller to localize user accounts, read user data, and ultimately authenticate users. You can use Samba's own net tool, but also Realm [1], as the front end for joining a client to a domain. With Realm, you need to specify the `--client-software=winbind` option to ensure that the Winbind service and not the system security services daemon (SSSD) is used to join a domain.

Cluttered Logfiles

The primary Winbind process creates a separate child process for each logical AD domain that the service wants to access. Each process is also assigned its own logfile, where you will find varying amounts of information depending on the configured logging level. If you experience issues with integration into a Windows environment, you should set the logging level to a high value to glean as much information as possible for debugging.

The problem in this case is that the sheer volume of log data makes it difficult to understand communication between the Winbind process and a domain controller. The individual entries each comprise a header and the message. Besides a timestamp, the header also contains various other details, such as the configured logging level, Winbind's process ID, the log message class, and the Winbind function that was used, as shown in the following example of a log message from the *nss_winbind* library:

```
[2023/05/04 16:20:51.998105, 3, ↩
  pid=1153814, effective(0, 0), ↩
  real(0, 0), class=winbind] ↩
  ../../source3/winbindd/winbindd.c:↩
  502(process_request_send)
process_request_send: ↩
  [nss_winbind (1153856)] ↩
  Handling async request: GETPWNAM
```

Because Winbind performs asynchronous function calls, the logfiles are difficult to read. The log message examples in Figure 1 illustrate this problem. The first entry belongs to a query for the user named *JOE* from AD domain *ADDOMAIN*, with the aim of discovering the user's security identifier (SID). The next message then outputs a SID. However, if you then use the `wbinfo` tool to validate the data, you will see that Joe has a SID ending in 1110 and the SID ending in 1107 belongs to Alice:

```
$ wbinfo --name-to-sid ADDOMAIN/joe
S-1-5-21-118052468-2300894008-↩
  1344842092-1110 SID_USER (1)
$ wbinfo --name-to-sid ADDOMAIN/alice
S-1-5-21-118052468-2300894008-↩
  1344842092-1107 SID_USER (1)
```

The problem now is simply that you cannot establish a relationship between the individual log messages because of the asynchronous function calls, which is precisely the problem the developers solved in Samba 4.17.

Winbind Tracing

To solve this relationship problem, the headers of the individual log messages have been extended to include a `traceid` field. In Figure 1, you can see this field and that the values for the field differ between the two messages. In the first case, the trace ID is 92, but in the second case it is 90. This output makes it immediately clear that

Lead Image © monsit jangariyawong, 123RF.com

the messages belong to different queries. If you are looking for the entries that belong to a specific query, you can now simply use a new tool, which has been around since Samba 4.19, to find the trace ID of log messages. Besides `traceid`, another new field, `depth`, lets you easily identify the nesting level of an individual request. This information is quite helpful, because a single query can generate many different sub-requests. The depth ID helps you identify the relationship between individual messages and the order in which the individual functions were called.

To identify the nesting level visually, Winbind even indents the sub-requests that belong to a trace ID by four spaces in each case. Thanks to this new type of Winbind log tracing, it is now very easy to identify the individual function calls that belong to a query so that you can clearly trace the communication paths.

Configuration and New Tool

To make sure Winbind uses these two new header fields in the log messages, you need to set a new option in the `/etc/samba/smb.conf` configuration file:

```
# winbind debug traceid = yes
```

To view all log messages that belong to a specific trace ID, simply use the new `samba-log-parser` tool:

```
# samba-log-parser ➘
  --traceid 92 /var/log/samba/
```

The beauty of the tool is that, to see all the messages of a trace ID and the file from which the messages originate, you only need to specify the ID and the log directory (Figure 2) – regardless of the logfile in which the message is stored. You can also use timestamps to sort the existing logs for a specific trace ID and generate a new logfile. As in the last example, `samba-log-parser` then shows the file from which the individual messages originate:

```
# samba-log-parser ➘
  --traceid 82 ➘
  --merge-by-timestamp /var/log/samba ➘
  > traces-82-by-timestamp.log
```

This option is helpful because you now have a chronological sequence of all messages for the specified trace ID in the `traces-82-by-timestamp.log` file. Finally, the tool also shows a flow tracing, which means you only get to see the functions, without any details, that a particular request uses. This output can be helpful for complex queries to establish the functions that are called without having to read the log messages. The `samba-log-parser` command then looks like:

```
# samba-log-parser ➘
  --traceid 81 ➘
  --flow /var/log/samba
```

Whereas the `traceid` and `depth` header fields are available regardless of the defined logging level, flow traces require a logging level of 20. You can easily set this by typing the

```
# smbcontrol all debug 20
```

command.

Conclusions

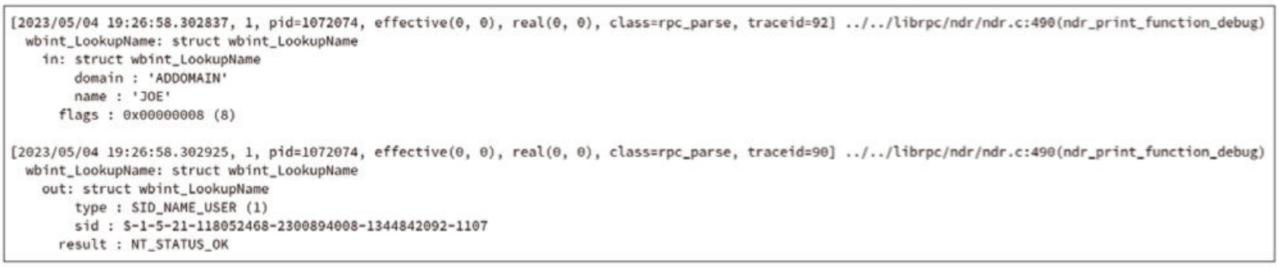
The new logging functions help you by making it easier to read the Winbind logfiles, giving you the ability to track down errors more quickly. Moreover, the `samba-log-parser` tool lets you view just the messages that are relevant to your use case. ■

Info

[1] Realm and Winbind: [https://www.freedesktop.org/software/realmd/docs/guide-active-directory-client.html#idm139657557854144]

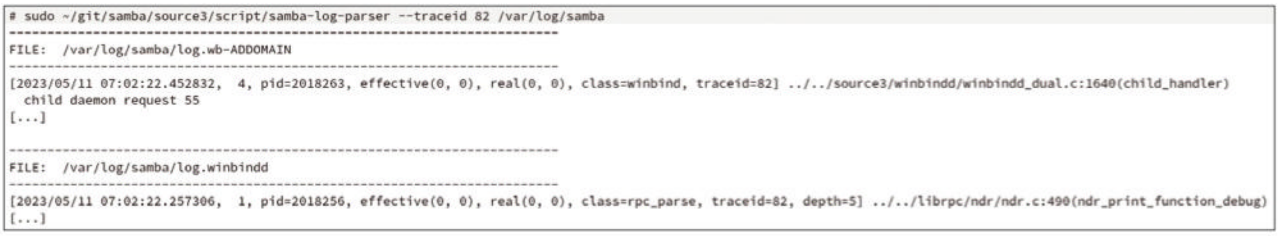
The Author

Thorsten Scherf is the global Product Lead for Identity Management and Platform Security in Red Hat's Product Experience group. He is a regular speaker at various international conferences and writes a lot about open source software.



```
[2023/05/04 19:26:58.302837, 1, pid=1072074, effective(0, 0), real(0, 0), class=rpc_parse, traceid=92] ../../librpc/ndr/ndr.c:490(ndr_print_function_debug)
  wbint_LookupName: struct wbint_LookupName
    in: struct wbint_LookupName
        domain : 'ADDOMAIN'
        name : 'JOE'
      flags : 0x00000008 (8)

[2023/05/04 19:26:58.302925, 1, pid=1072074, effective(0, 0), real(0, 0), class=rpc_parse, traceid=90] ../../librpc/ndr/ndr.c:490(ndr_print_function_debug)
  wbint_LookupName: struct wbint_LookupName
    out: struct wbint_LookupName
        type : SID_NAME_USER (1)
        sid : S-1-5-21-118052468-2300894008-1344842092-1107
      result : NT_STATUS_OK
```

Figure 1: Because of Winbind's asynchronous function calls, logfiles cannot be read sequentially.



```
# sudo ~/git/samba/source3/script/samba-log-parser --traceid 82 /var/log/samba
-----------------------------------------------------------------------
FILE:  /var/log/samba/log.wb-ADDOMAIN
-----------------------------------------------------------------------
[2023/05/11 07:02:22.452832,  4, pid=2018263, effective(0, 0), real(0, 0), class=winbind, traceid=82] ../../source3/winbindd/winbindd_dual.c:1640(child_handler)
  child daemon request 55
[...]

-----------------------------------------------------------------------
FILE:  /var/log/samba/log.winbindd
-----------------------------------------------------------------------
[2023/05/11 07:02:22.257306,  1, pid=2018256, effective(0, 0), real(0, 0), class=rpc_parse, traceid=82, depth=5] ../../librpc/ndr/ndr.c:490(ndr_print_function_debug)
[...]
```

Figure 2: The new `samba-log-parser` tool shows all messages from different logfiles for a trace ID.

Zero trust planning and implementation

Take Your Mark

The many facets of the zero trust implementation process can be a source of frustration, which is why we offer a step-by-step guide to implementing zero trust models to help you make state-of-the-art IT security become a reality. By Martin Kuppinger

The zero trust model was published in 2010 by John Kindervag, who was employed by IT analysts Forrester Research at the time. However, the foundations for zero trust were laid down as early as 1994 by Stephen Paul Marsh in his doctoral thesis at the University of Stirling (Scotland). The strategy only really became popular in 2020 when, as a result of the coronavirus pandemic, many companies had to switch to home offices and new labor models at short notice, putting their previous safety solutions to the test. As a result, many companies defined zero trust as the core of their cybersecurity setups and launched projects to match.

The steps from the basic model to a concrete implementation are painstaking, partly because the model was initially very network-centric (zero trust networks) and primarily postulated generic requirements. However, the zero trust architecture [1] from the US National Institute of Standards and Technology (NIST) and a position paper from Germany's BSI [2] (for which the institute expressly invites suggestions, comments, and criticism) have since been released.

Basic Principles

Zero trust originally focused on security in network infrastructures, with the focus on preventing lateral movement (i.e., preventing attackers from moving relatively freely on the network to attack systems after working around the firewall). The next basic idea was not trusting individual components, but rather carrying out continuous verification at different points and on different levels: never trust, always verify. The fundamental cornerstones of zero trust are derived from:

- Continuous verification or, to be more precise, repeated verification of users, devices, and applications during access and in ongoing sessions, because they are all considered inherently insecure
- The minimum principle, which is the assignment of only the minimum required authorizations that, ideally, should also be assigned just-in-time (JIT; i.e., only as soon as and while they are needed)
- Multilevel security, with checks carried out repeatedly and at different levels

Also cited frequently is microsegmentation of networks to reduce attack surfaces, create more points for verification, and contain the spread of attacks. However, this measure is just one technical implementation among many.

Figure 1 provides a slightly different and broader view: Users rely on their devices to access services on the network. In turn, services are at the system and application levels to access and manipulate data, and software forms the basis. Access is also supported for service accounts and nonhuman identities, but the principle remains the same. On the one hand, the figure illustrates the complexity of zero trust as a strategy, because it does not just apply in a single place. On the other hand, it shows how it is possible to break down the comprehensive model into smaller parts and focus on a variety of security measures across the entire chain, which then come together to form a zero trust approach.

Important Technical Components

Figure 1 shows the areas in which measures are focused when establishing zero trust security:

- Identities with identity and access management (IAM) and especially multifactor authentication (MFA), which ideally no longer works with passwords (passwordless authentication)

Lead Image © magiceyes, 123RF.com

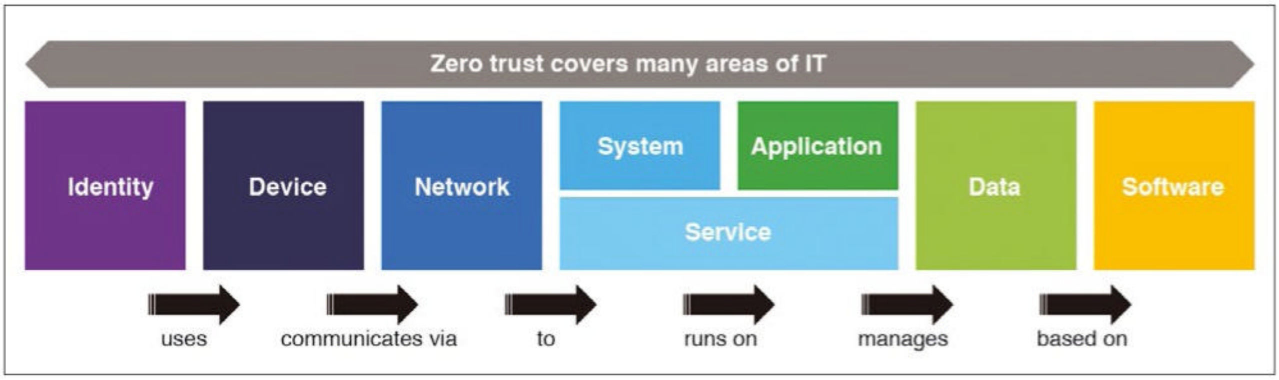


Figure 1: Zero trust encompasses many areas of IT, so successful approaches require numerous measures.

- Devices with terminal device security and device management
- Networks with microsegmentation and zero trust network access (ZTNA)
- Systems with hardening and access control in conjunction with IAM.
- Applications with hardening, access control by IAM, and, if possible, dynamic authorization by policy-based access management (PBAM)
- Data security and data governance
- Software with supply chain security

Figure 2 shows additional interdisciplinary functions. Of note are policy management and monitoring in the broadest sense (i.e., topics such as extended detection and response, XDR). Other aspects such as incident response management; security information and event management (SIEM) and security orchestration, automation, and response (SOAR) tools as part of monitoring; and attack surface management are also key components of a zero trust solution.

Policies or guidelines, which NIST also emphasizes in its zero trust architecture, are important elements. Security must be controlled by policies, which has long been the case at various levels, such as firewall policies for network access. What is still missing, but is not an obstacle to implementing zero trust models, are consistent solutions at all levels (e.g., PBAM in the application access authorization area) and consistent tools for policy management and governance. Consistent policy governance can also be implemented without technical tools for cross-system policy management.

Ultimately, these very large numbers of fields of action are the result of the many facets of cybersecurity that cannot be reduced to individual technologies. On the other hand, organizations do not typically need to start from scratch, but already have various elements in place. It is therefore usually more a question of completing and optimizing rather than completely rebuilding cybersecurity.

Organizing and Planning

The key to success with zero trust is easy to define and comprises two aspects. The first is reducing complexity, and the easiest way to resolve this problem is to break it down into small

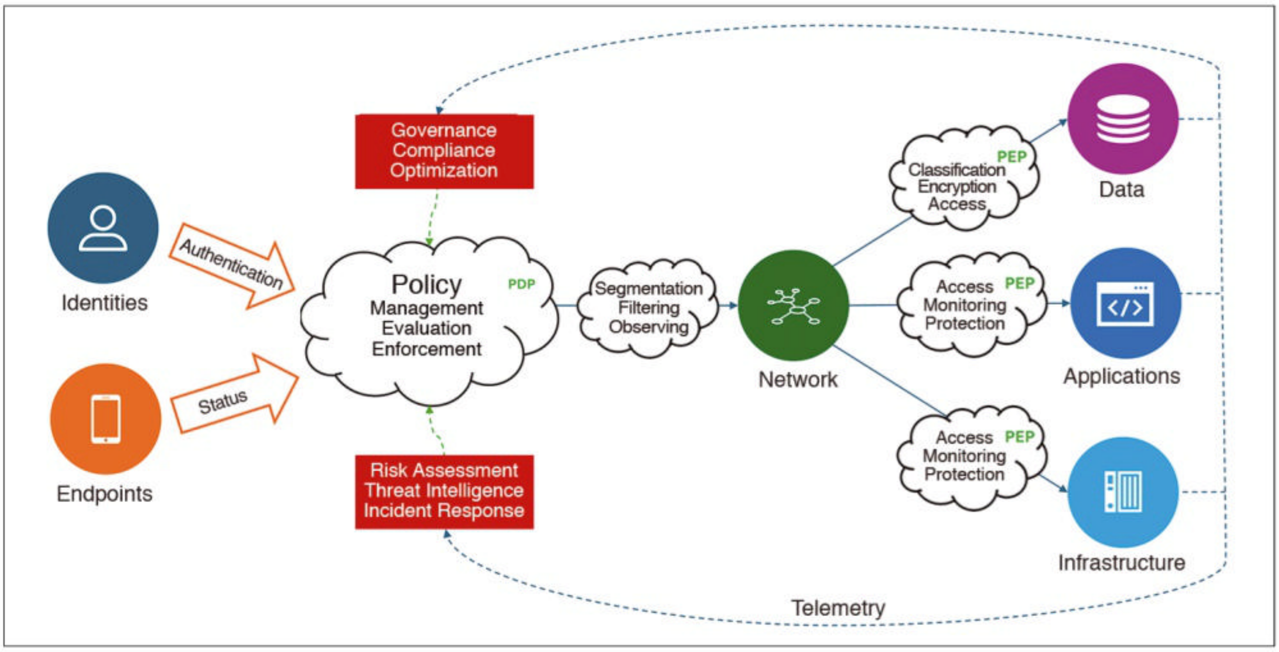


Figure 2: Zero trust relies on policy-based control of access across different units and on continuous monitoring.

parts, which IT managers then look at individually – but without losing sight of the big picture – before putting everything back together. Figure 1 shows that zero trust comprises many elements that can be resolved separately. In other words, the aim is to define and implement manageable projects in a zero trust program while defining the superstructure (e.g., overarching monitoring and policy governance).

The second aspect is concretization. Although zero trust is an abstract concept, the individual elements can easily be concretized. What do you need to improve network security? What does modern IAM look like? For these simple, clear-cut questions, the basic idea behind zero trust (i.e., never trust, always verify) acts as an acid test. You need to ask yourself whether the chosen approach is one of many layers in a multilevel security model with continuous monitoring or whether IT is just creating another layer that it trusts.

The first step before defining projects in a program is to understand what needs to be done. The best starting point is a business impact analysis (BIA) to help you understand which parts of business operations and the underlying IT and operational technology (OT; e.g., in production) are particularly critical, which together with attack surface management (ASM) makes it possible to determine where criticality for business operations and risks from attack surfaces are particularly high. On this basis and armed with the knowledge of the basic building blocks in a zero trust approach, a rough ideal scenario can also be sketched out that lists the necessary elements and can in turn be mapped against the assessed risks and critical areas.

The next organizational step is gap analysis, a critical comparison of the required and prioritized components and the current status. This step helps to identify the biggest gaps and subsequently to develop a roadmap to further improve the current state of the cybersecurity infrastructure in line with zero trust. The result is the logical sequence of a concrete program and project planning.

You should always keep critical comparison in mind. Unfortunately, defensive arguments are often used in gap analyses because individual departments and managers feel attacked when gaps and the need for change are brought up. However, the objective is to take the next step toward state-of-the-art security strategies and technologies – not to criticize what has been achieved so far.

When progressing toward a zero trust model, it is also important to rethink the organization. As a rule, this part is the responsibility of the person in charge of IT security or the chief information security officer (CISO). In practice, however, this is not always the case for all elements. In some organizations, IAM is still assigned to the IT infrastructure and not to the CISO. Network security is sometimes the responsibility of a separate network and communications division; application security is often organized in a decentralized way; and terminal device security is often the task of client management. Ideally, companies would want to adapt to ensure that all security-related issues are dealt with by the CISO. If this arrangement is not feasible, the other departments must be involved in the zero trust project, and the CISO must receive the necessary backing from IT management and the executive board.

The Correct Starting Point

One frequently asked question is, "Where should the journey to zero trust start?" Strictly speaking, the answer can be found in the systematic approach outlined above. The risk and gap analyses clearly show the most critical topics. IT teams need to prioritize these issues within the framework of budget availability, but at least one area is always going to be a good choice.

As Figure 1 shows, zero trust starts with identity and authentication. If not yet implemented, MFA is always going to be a good place to get the ball rolling because it is a central element of zero trust. The same applies to the development and expansion or modernization of IAM, which includes not only authentication, but also the management of all types of identities and user accounts, the control of access authorizations, and PBAM for dynamic access authorization.

On the other hand, you should also understand that no single solution will allow you to make zero trust architectures a reality. Even approaches such as ZTNA, where "zero trust" is part of the name, are only partial elements in this kind of solution. A differentiated assessment must be made as to whether and to what extent these elements are necessary.

Other sub-elements such as microsegmentation are important, as well, but by no means always necessary. For organizations that work with flexible working models and access from different locations, but only use cloud services and do not have internal IT, microsegmentation is irrelevant; however, it does play a role if many IT services are still operated internally in data centers.

Conclusions

Ultimately, besides IAM and MFA as the logical technical starting points, the correct entry point for zero trust is solution-oriented work, which helps you develop concrete and programmatic planning from an abstract, complex, and often diffuse strategy and implement the planning, step by step.

One thing is clear: Zero trust is not outdated, but a model that will continue to shape cybersecurity in the coming years in concrete implementations long after the hype has disappeared. ■

Info

[1] NIST zero trust architecture: [https://www.nist.gov/publications/zero-trust-architecture]

[2] BSI position paper: [https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeLeitlinien/Zero-Trust/Zero-Trust_04072023.pdf?__blob=publicationFile&v=4] (in German)

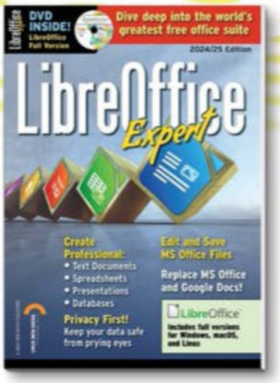
Author

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

Sovereign Cloud Stack
Copilot
RaMP
ADMIN
Network & Security
ISSUE 82
Sovereign Cloud Stack
Digital freedom, both foreign and domestic
Emulating and Mocking AWS and GCP Services
Rapid Modernization Plan Securing Active Directory
Legal or Litigation Holds and eDiscovery in Microsoft Teams
Copilot: Control or block GPT-4-based AI chat
Go Testing Frameworks
Bacula
Flexible backup for large environments
AIX Automation
IBM AIX collections from Ansible Galaxy
Topgrade
Update and upgrade your Linux apps
Spacelift
IaC management platform
KALI LINUX
DVD INSIDE
LINUX NEW MEDIA
WWW.ADMIN-MAGAZINE.COM

LINUX MAGAZINE
Ollama
Run advanced AI models on your home computer
10 GLITTERING GEMS FROM THE FOSS VAULT

FIND THE PERFECT TOOL FOR THE TASK
Cool Linux Hacks
Tricks and shortcuts for Linux geeks
114 HACKS



Dive deep into the world's greatest free office suite
2024/25 Edition
LibreOffice Expert

RASP PI TRICKS
MakerSpace #04
HANDS-ON PROJECTS FOR MAKERS

Manage status messages in CouchDB with MapReduce

On the Couch

CouchDB offers numerous interesting features for acquisition and filtering of status messages that make it a fast and convenient data storage solution. By Oliver Kurowski

Whether the Internet of Things (IoT) or a server landscape, microservices or cron jobs, applications produce all kinds of status messages that you need to collect and evaluate. Whether it's an alert from the robot lawnmower, the abrupt termination of a long-running task, or simply a weather warning, storing the various messages centrally and evaluating them independent of their structure is always going to be a challenge (Listing 1).

Listing 1: Example Status Messages

```
{ "timestamp":"202405141201",
  "source": "gardenrobot-1",
  "message": { "type":"alert",
               "value":"animal" }
}
{ "timestamp":"202405141220",
  "source": "gardenrobot-1",
  "message": { "type":"warning",
               "value":"low power" }
}
{ "timestamp":"202405150200",
  "source": "server"
  "message": { "type":"task done",
               "value":"night backup",
               "result":"done",
               "errors":[] }
}
{ "timestamp":"202405160800",
  "source":"18739949083333",
  "rss":"weatherchannel",
  "region":"Berlin",
  "message": { "type":"warning",
               "value":"rain",
               "category":"heavy",
               "chance":"80%" }
}
```

CouchDB can help with centralized acquisition and subsequent filtering of status messages (e.g., by number, hour, or source). Its main advantage, and one that it shares with other NoSQL databases, lies in the schema-free nature of the data. Each CouchDB dataset can have its own structure, as long as it can be mapped in JSON format. This freedom means that many different status messages can be managed and queried in a single database without the need for tables adapted to the structure of the messages.

If the format of a status message changes, you just need to reflect this in the queries; no changes need to be made to the database that stores the messages. A document-based database of this type makes sense for dissimilar data structures such as status messages from different sources.

If the database – like CouchDB – also supports simple clustering, replication, and query options, it is definitely worth a second look.

CouchDB is one of the original NoSQL databases. Damien Katz developed the software back in 2005. The basic idea came from his previous job as a senior developer at Lotus Notes, distributed collaboration software. He combined the schemaless, document-oriented approach of Lotus Notes with the – at the time – relatively new MapReduce technology, which can query large amounts of data on distributed systems.

CouchDB has been an Apache project since 2008. Version 1.0 in 2010 has evolved into version 3.3 today. "Couch" was originally an acronym for "cluster of unreliable commodity hardware," which reflected the fact that the system also works well without powerful high-availability servers. However, a second meaning (represented by the logo with the couch) could refer to the simplicity with which databases can be set up without a fixed schema.

The capabilities of CouchDB go beyond a pure key-value store. The database system shines with a multimaster replication model, ACID-compliant (atomicity, consistency, isolation, and durability) document

Photo by Austin Distel on Unsplash

storage, indexing functions or MapReduce technology in JavaScript, and the Mango query language.

The needed binaries and information for installing CouchDB can be found on the project's website [1]. For Windows and macOS, just download the installers directly; for CentOS/Debian and Ubuntu, a little typing at the command line is all it takes to install the packages directly from the repositories, where the source files also reside.

CouchDB is written in Erlang/OTP (Open Telecom Platform), a functional language from the world of telecommunications. The strengths of this language type are its simple parallel processing, fault tolerance, and robustness. From the outset, CouchDB development focused on distributed databases on the network. Erlang was the tool of choice for this application to ensure specifically the security and consistency of data in a cluster with a high load.

During installation, you need to distinguish between a standalone install and a variant as part of a cluster; the standalone installation is fine if you just want to gather an initial impression.

Fauxton

CouchDB does not use its own transport format for communication but relies entirely on its HTTP REST API. After completing the installation with the default values, the CouchDB instance listens on port 5984. Depending on the installation (local host or IP address or domain name), with a call in your browser or a GET request sent to the base address, you can request a short status message as a test:

```
{ "couchdb":"Welcome", "version":"3.3.3", ↩
  "git_sha":"40afbcfc7", "uuid":↩
  "3b74c04721ee61dbe9db74ac3c69e8f8", ↩
  "features":↩
  ["access-ready", "partitioned", ↩
   "pluggable-storage-engines", ↩
   "reshard", "scheduler"], ↩
  "vendor":{"name":"The Apache Software ↩
                   Foundation"} ↩
}
```

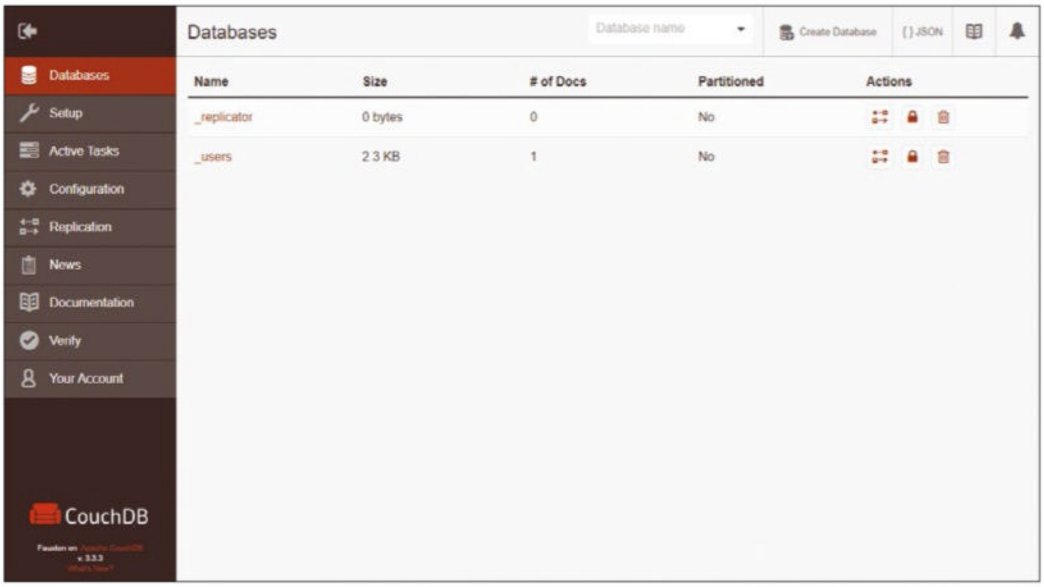
The vendor name can be changed easily later on, as can the port and other settings that have not been addressed here. The built-in front end, Fauxton, is fine for getting to know the basics of CouchDB. I'll assume you have a local installation and can reach Fauxton on *http://localhost:5984/utils*. Initially, you will see a database overview (Figure 1).

The two internal databases *_replicator* and *_users* are already in place. As shown in Figure 1, the *Databases* choice in the sidebar shows all existing databases and their details. You can also adjust the security settings here for each database and delete databases. Once a database has been created, it cannot be renamed or automatically emptied. The installation type can be found in *Setup*, with the choice of *Configure Single Node* or *Configure Cluster*. The *Active Tasks* item takes you to all active tasks in CouchDB. *Configuration* is where can you adjust some of the settings and add some extra, non-standard entries to the settings. *Replication* takes you to a list of current and past replications. You can also create a new replication at this point. Selecting *News* lets you integrate news from a blog, *Documentation* contains links to various documentation sources, *Verify* lets you check the installation, and *Your Account* is where you manage the current admin account or set up new admin accounts.

Storing Messages

The question now is how the status messages get into CouchDB, for which no special CouchDB format or protocol exists. All actions rely on the HTTP REST standard. Whether browser, Python, `curl`, Postman, or Lisp – anything that speaks HTTP REST can be used. Of course, many languages have helpers and wrappers that translate more complex tasks into the HTTP calls, but – at the core – all actions are `GET`, `PUT`, `POST`, or `DELETE` calls, and on port 5984 by default. After the install, you can change the port if needed.

The hierarchy of a CouchDB installation is relatively flat, starting with databases, and internally each database stores JSON documents as logical and physical units. A document can contain additional data and binary non-JSON formats as attachments.



Name	Size	# of Docs	Partitioned	Actions
_replicator	0 bytes	0	No	
_users	2.3 KB	1	No	

Figure 1: The Fauxton interface initially shows a database overview.

Listing 2: Command-Line Communication

```
### Create a database
$ curl -X PUT localhost:5984/messages -H "Authorization: Basic YWRtaW46YWRtaW4="

### Delete a database
$ curl -X DELETE localhost:5984/messages -H "Authorization: Basic YWRtaW46YWRtaW4="

### Show details of a database
$ curl -X GET localhost:5984/messages

### New document with PUT
$ curl -X PUT http://localhost:5984/messages/first_message -d '{"message":"Hello"}'

### New document with POST
$ curl -X POST http://localhost:5984/messages -d '{"_id":"second_message","message":"World"}' -H
  "Content-type: application/json"

### Retrieve document with GET
$ curl -X GET http://localhost:5984/messages/first_message

### Retrieve all documents in a database
$ curl -X GET http://localhost:5984/messages/_all_docs

### Retrieve selected documents including content
$ curl -X POST http://localhost:5984/messages/_all_docs?include_docs=true -d '{"keys":["first_
  message","second_message"]}' -H "Content-type: application/json"
```

The names of everything that is important for the CouchDB system itself start with an underscore – be it the system databases *_users* and *_replicator*, documents (e.g., `_design`), or document fields (e.g., `_id` and `_rev`). The user cannot create databases or documents and fields with a leading underscore unless they belong to the CouchDB system.

A username and admin password were already entered during the installation. For the sake of simplicity, this account is also used in Fauxton and for creating databases with `curl` or Python. If you have a username/password combination of `admin/admin`, the attribute as used in the calls is

```
Authorization: Basic YWRtaW46YWRtaW4=
```

Listing 3: Creating a Database in Python

```
import json
import urllib3
http        = urllib3.PoolManager()
COUCHDB_URL = "http://localhost:5984"
AUTH        = 'Basic YWRtaW46YWRtaW4=' # admin/admin
HEADERS     = {'Content-type': 'application/json','Authorization': AUTH}
def create_database (database_name):
  url    = f'{COUCHDB_URL}/{database_name}'
  result = http.request('PUT', url, headers=HEADERS)
  return (json.loads(result.data))
print(create_database ("messages"))
```

Of course, you will want to create different users and roles in production operation. Just for the sake of completeness, it should be mentioned that you can automatically check (`validate_doc_update` function) or change (`update` function) the data when saving documents.

Databases

Starting with the Database view in Fauxton, create a database named *messages* by clicking the *Create Database* button at top. The *Non-partitioned* setting is fine here. Very large databases can be partitioned if you create them such that queries are only ever made against a specific subset of the data. After creating a database in Fauxton, you are taken directly to the Database view. The command-line alternatives to the actions described for Fauxton are shown in Listing 2. The response from CouchDB is a short *{ok:true}* if successful or an error message if not:

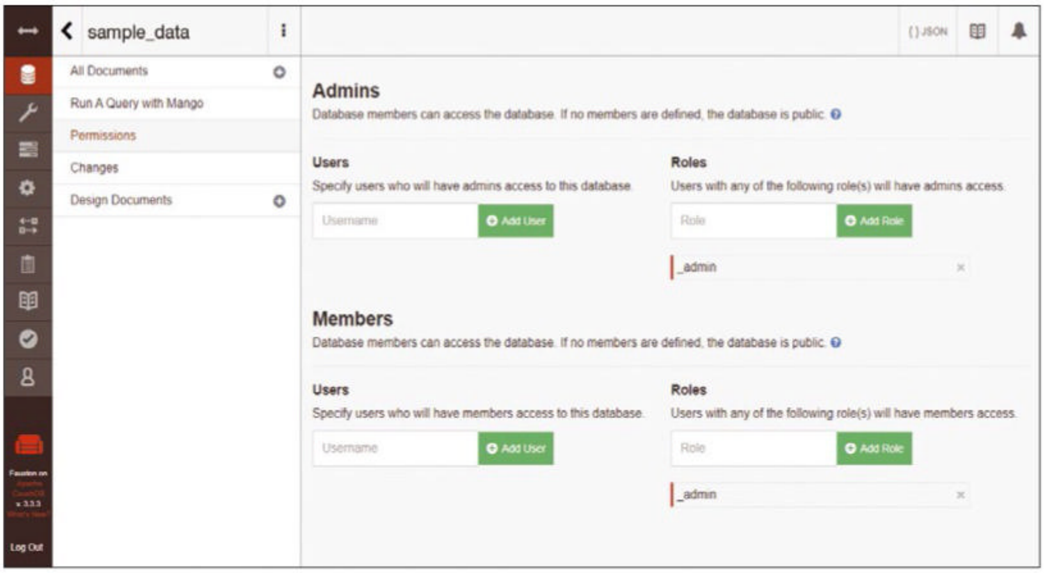
```
{"error":"file_exists","reason":➘
 "The database could not be created, ➘
  the file already exists."➘
}
```

Communication with CouchDB is most likely going to be through an application. In the basic structure, a program must send `GET`, `PUT`, `POST`, and `DELETE` requests and populate the URL, the authentication headers, and the data to be transferred. Everything else is controlled by the content of the requests. Listing 3 shows a simple example in Python: a basic program that creates a database and outputs the result of this operation. The database is created on the first run, a second run would provoke an error message.

The CouchDB security strategy stipulates that access to a database must be governed by means of users or roles. In the Fauxton database overview, you can set the access rights for each database in the Actions column (Figure 2). Creating a database automatically defines the logged-in user (*admin* in this example) as the admin. If you remove all users and roles under the *Permissions* item of a database, you end up with a public database anyone can read and write without authentication, which is fine for an initial test in a local environment. Authentication headers are no longer required for the requests, which certainly saves some typing when you are trying out `curl` at the command line.

Just as easily as the database can be created, it can be deleted again with a `DELETE` instead of a `PUT` request (Listing 2), but be careful: The delete action happens immediately. All data in the database is lost. In Fauxton, you can delete a database in the Actions column of the database overview. Fauxton prompts you before deleting, just to be on the safe side.

A database in a CouchDB installation encapsulates data and queries. With one exception, CouchDB has no real joins, so you do not need to consider whether documents in *messages* can



Admins

Database members can access the database. If no members are defined, the database is public.

Users

Specify users who will have admins access to this database

Roles

Users with any of the following role(s) will have admins access

_admin

Members

Database members can access the database. If no members are defined, the database is public.

Users

Specify users who will have members access to this database

Roles

Users with any of the following role(s) will have members access

_admin

Figure 2: Configuring users and roles in the Permissions view in the Fauxton graphical user interface.

be linked to documents in another database. Within a database, each document has a unique ID, but the same ID can also occur in other databases. Once created, some database information and statistics can be retrieved by `GET` requests. Please remember, authentication is not necessary if no users and no roles are defined for this database.

Documents

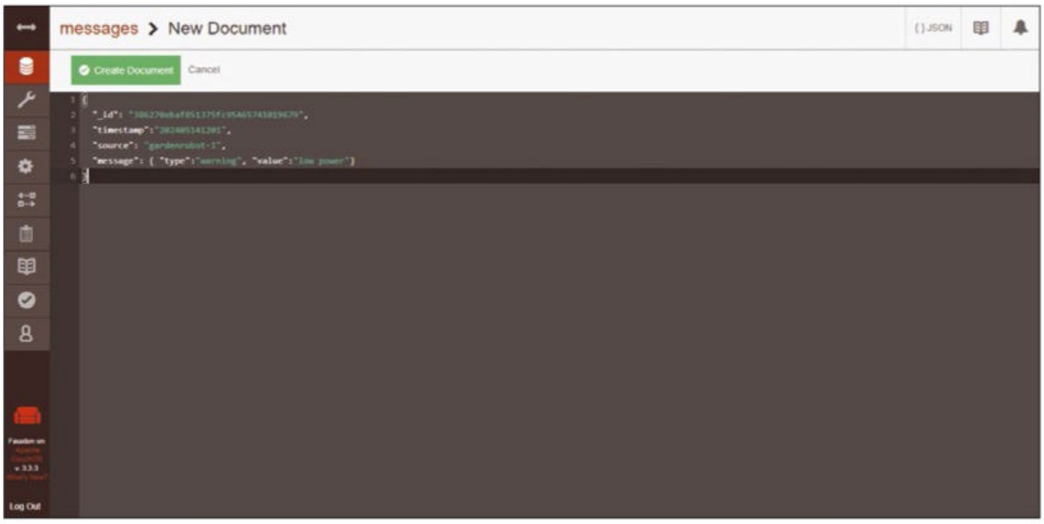
To save, you can either use a `PUT` request with the ID as part of the URL or a `POST` request to the desired database. A `POST` request with a new document must contain the `_id` field. If it is missing, or if the ID of the document is irrelevant, you can leave the field blank when creating the document. CouchDB then assigns a universally unique ID (UUID) for the new document. `POST` requests to CouchDB have a `Content-type:application/json` header. After successfully storing a document, CouchDB returns the ID and the initial revision number of the new document. In the event of an error, the database outputs a message to that effect:

```
{"ok":true,"id":"second_message","rev":⮐
 "1-183d19dc77574e297dc791f3723caf41"}
```

Even in a program, saving a document with `PUT` or `POST` does not pose any major challenges. If you don't feel like using the command line or writing code, you can use Fauxton to create a document with valid JSON by clicking *Create Document* in the desired database. CouchDB then suggests a UUID as the `_id`; you can change this value before saving, if so desired.

Now is a good time to store the list of status messages from Listing 1 in the *messages* database with one of the three options (Figure 3).

One more thought regarding the selected IDs of the documents: If sorting by time or other ascending values is an important criterion, an ID with a leading timestamp is recommended. Without programming a query, you can limit the time range with CouchDB's built-in resources by adding the appropriate options `startkey` and `endkey` to the internal `_all_docs` query. (I get back to this later.) However, it is not always possible to guarantee uniform IDs, especially with many status messages arriving from different systems, which is why the format of the ID is not taken into account in this example; you will want either to use `POST` to store the messages without specifying a `_id` field or to adopt the CouchDB UUID from Fauxton after running *Create Document*.



```
{
  "_id": "386270ebaf851375fc95465741819676",
  "timestamp":"202405141201",
  "source": "gardenrobot-1",
  "message": { "type":"warning", "value":"low power"}
}
```

Figure 3: When creating a document in Fauxton, always consider whether an ID preceded by a timestamp is a good choice.

ACID, Append Only, MVCC

The revision number, which CouchDB automatically inserts into the document when it is first saved, has a central function in the CouchDB system. If you don't specify the revision number, which comprises a consecutive version number and a hash of the content, documents can be neither changed nor deleted. Moreover, without a specified revision number you have no update options. Why is that?

Data is stored in CouchDB by appending newer data to older data. It fulfills the ACID conditions. A data record is either written completely or – in the event of an error – not written at all. Changes to a data record are appended to newer versions of a document, thus eliminating the need to lock data records during write operations. As long as you do not explicitly delete the data of the old revisions, you can retrieve the entire history of a document with old revision numbers.

To manage concurrent data changes, CouchDB uses the multiversion concurrency control (MVCC) approach, which relies on the revision number. This property plays an important role, especially when several read/write operations take place in very quick succession. If two users open and modify document A with revision number *1-c6438911bbf* in one program, the first program that enters the previously valid revision number when saving wins. The save action increments the document's revision number. The version that is backed up second now has an outdated revision key, which means a document update conflict error is reported.

At the end of the day, the user program is responsible for conflict management.

The situation is different with CouchDB replications. If the same document is changed in a cluster and replication is delayed (e.g., because of faults), the database system decides which version wins and keeps the second document as the previous version.

MapReduce and Views

Without its own queries, CouchDB is initially just a key-value store with the option of delimiting the document set on the basis of IDs. Individual documents are read with a `GET` request that specifies the database name and the document ID. To retrieve multiple documents, CouchDB's own `_all_docs` query is used at the database level.

This query already anticipates the opportunities that MapReduce offers. The `_all_docs` function returns a list of all documents in a database, including the current revision numbers. What is missing, though, is the content of the documents. If you add the `include_docs=true` parameter to the URL of the query, CouchDB also outputs the document data in the `doc` entry in each line.

To narrow down the list of desired documents, you can use the `_all_docs` query with `startkey=<xxx>` and `endkey=<yyyy>` to define the range of keys within which you want to receive the documents. However, these queries only make sense if the document IDs are structured such that they can be easily narrowed down (e.g., by specifying the time).

Listing 4: Design Document with Views

```
{
  "_id": "_design/queries",
  "_rev": "6-856a5c52b1a9f33e136b7f044b14a8e6",
  "language": "javascript",
  "views": {
    "by-timestamp": {
      "map": "function (doc) {\n  if (doc.timestamp) {\n    emit(doc.timestamp, null);\n  }\n}"
    },
    "by-hours": {
      "map": "function (doc) {\n  if (doc.timestamp) {\n    emit(doc.timestamp.substr(8,2), null);\n  }\n}"
    }
  }
}
# Result of a query /messages/_design/queries/_view/by-timestamp
{"total_rows":3,"offset":0,"rows":[
{"id":"386270ebaf851375fc95465741019679","key":"202405141201","value":1},
{"id":"386270ebaf851375fc9546574101a657","key":"202405150200","value":1},
{"id":"386270ebaf851375fc95465741020597","key":"202405160800","value":1}
]}
```

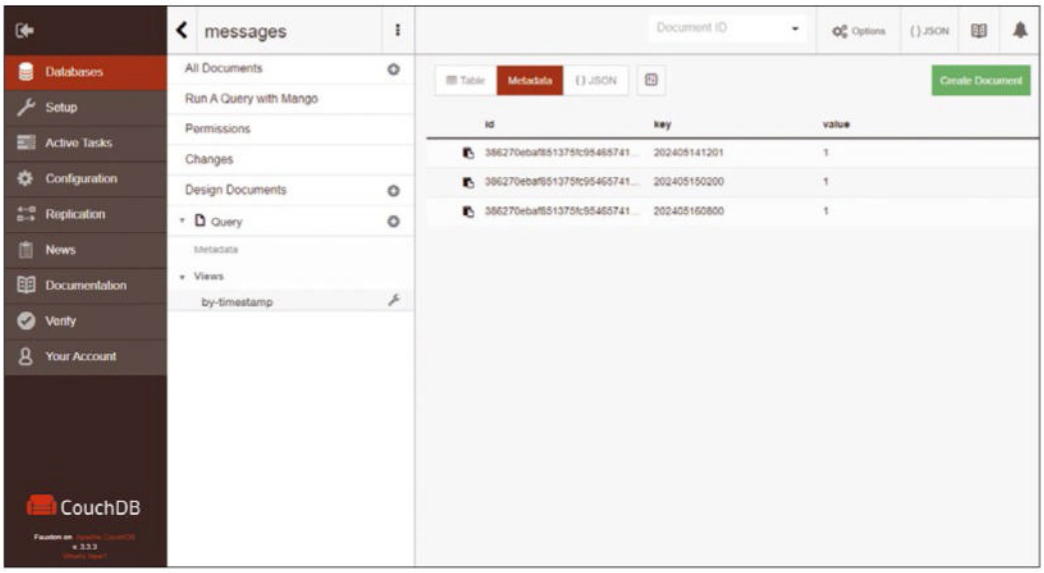
If you want the database to load a few very specific documents, send an `_all_docs` POST request with the `keys":[<key1>,<key2>,...]` attributes. Again, the document content is only returned if you stipulate `include_docs=true`. However, the capabilities of CouchDB are by no means limited to saving documents schematically and retrieving them again by specifying the ID.

For example, perhaps you only want status messages that include an alert to appear or just all status messages of a certain type. How can you implement this if you can only search for the ID of a status message and these IDs are also random UUIDs? CouchDB is not a plain-vanilla key-value store but can use MapReduce indexes to create queries other than by ID. The standard approach is to program your own `map` and `reduce` methods in JavaScript. Other languages can be implemented, as well, by changing the query server setting of the instance. Internal reduce functions such as `_count` or `_sum` are implemented natively in Erlang and are therefore very powerful during indexing.

An important point with MapReduce is that you have no access to other documents during indexing. Each document is considered separately. However, you can embed a referenced document in the result at query time with `_include_docs=true`. The `map` function always expects a complete document as an input parameter. Depending on the desired index, the `emit(key,value)` command writes an entry to the index:

```
function (doc) { emit(doc.timestamp, 1); }
```

The `value` can be freely selected and can be zero. Note that, if you want to group the list later with a `reduce` function, you will need a `value` that can be totaled or counted. Later, you will want statistics on the number of messages per hour, so first add a 1 as



id	key	value
386270ebaf851375fc95465741…	202405141201	1
386270ebaf851375fc95465741…	202405150200	1
386270ebaf851375fc95465741…	202405160800	1

Figure 4: You can view the result of an index update in Fauxton.

a value to each key. The reduce step is optional; it is necessary if you want to, say, total the results of the list or group the query. To help you get started, one of the natively integrated reduce functions (e.g., `_sum` or `_count`) is a good choice. A MapReduce function that writes an index and optionally aggregates it is known as a view in the CouchDB world.

Design Documents

Queries and other functions are stored in documents in the database in which the data documents are also stored. The document ID always starts with `_design/<name>`, which is why it is known as a design document. Not all queries in a database have to be saved in a single design document; on the contrary, it makes sense to distribute them across different design documents for performance reasons. You can imagine a design document as being something like a container for a number of CouchDB functions (view, update, filter, validate). Right now, the MapReduce functions stored as views in the design document are of interest. To retrieve a list of all timestamps for the status messages, you need to save a design document with the `views` and the `map` function in the *messages* database (Listing 4).

Because design documents are completely normal data documents, apart from the special ID, the save processes are also the same. Whether you choose `PUT`/`POST` or you use Fauxton, you just add the design document to the database as usual. Note that you must replace the line endings of the functions with newlines (`\n`) when you save the design document.

After saving or modifying a design document with views, indexing all documents stored in this database starts in all of the design document's views. Depending on the number of documents, this process can take some time. You can monitor the progress in Fauxton under *Active Tasks* or by sending a `GET` request (with authentication) to *http://localhost:5984/_active_tasks*. If you add new data documents later on or modify a data document, this function is automatically called up for this one document. The call updates the existing index.

The result of a query can be viewed directly in Fauxton (Figure 4).

The results are a little more compact if you use a `curl` query or simply point the browser at *http://localhost:5984/messages/_design/queries/_view/by-timestamp*. In the query results, the desired index (*timestamp*) is now defined as the key with sorting in ascending order. If the index has documents, each view provides the ID of the document and the key-value from the `emit` statement.

Of course, you do not want to receive all the list data as a result. Entering the keys in an internal B+ tree lets you quickly implement the desired subset queries. The `startkey="20240515"` parameter shows all status messages from May 15, 2024, when you call the view, whereas the `endkey="20240516"` parameter limits the view range to messages before May 16 of that year. A time range can be limited by combining `startkey` and `endkey` in one call. Bear in mind that the end key is part of the result for a standard query. Accordingly, a query of

```
startkey="20240516"&endkey="20240516"
```

returns no results because the search starts on and only returns results up to `20240516`.

The standard query returns a list of IDs and keys, but not the contents of the documents. Either the user program can download the documents individually from the list of IDs, or you need to specify the `include_docs=true` parameter in the query. As with `_all_docs`, the data document can then be found in the output list below `doc` in each line.

The example also reveals the limits of queries with mapping. This query does not reveal which messages occurred in the morning or at night. Subset searches only work from front to back. To search for the time, you need a second view that only outputs the hour (Listing 5).

Reduce Function

Because the topic of reduce functions is relatively extensive [2], I will only look at CouchDB's own `_count` function here. Once a reduce function

Listing 5: Views by Timestamp and Hour

```
{
  "_id": "_design/queries",
  "_rev": "6-856a5c52b1a9f33e136b7f044b14a8e6",
  "language": "javascript",
  "views": {
    "by-timestamp": {
      "map": "function (doc) {
        if (doc.timestamp) { emit(doc.timestamp, null); }
      }"
    },
    "by-hour": {
      "map": "function (doc) {
        if (doc.timestamp) { emit(doc.timestamp.substr(8,2), null); }
      }"
    },
    "by-hour-count": {
      "map": "function (doc) {
        if (doc.timestamp) { emit(doc.timestamp.substr(8,2), 1); }
      }",
      "reduce": "_count"
    },
    "by-source-type-count": {
      "reduce": "_count",
      "map": "function (doc) {
        if (doc.source && doc.message.type && doc.message.value) {
          emit([doc.source, doc.message.type,doc.message.value], 1);
        }
      }"
    }
  }
}
```

Listing 6: Reduce Without Grouping

```
$ curl -X GET localhost:5984/messages/_design/queries/_view/by-hours-count
{"rows":[
{"key":null,"value":4}
]}
```

Listing 7: View Grouped by Key

```
$ curl -X GET localhost:5984/messages/_design/queries/_view/
by-hours-count?group=true
{"rows":[
{"key":"02","value":1},
{"key":"08","value":1},
{"key":"12","value":2}
]}
```

is available in a view, the mapping results are no longer output as a list, but totaled by the reduce function.

To evaluate the number of status messages per hour, you need to modify the `by-hour` view slightly (**Listing 5**). A 1 is written to the index as a separator for later counting and summarizing. A normal call of the view first totals all rows and outputs the sum as the result. In this case, the `key` field is `null` (**Listing 6**), but this result isn't exactly the one expected. The secret lies in the `group=true` parameter in the query, which specifies that the total output be sorted by `key`, and returns the desired results (**Listing 7**).

Now not only simple values can be used as keys, but also arrays, so that a single view can group and count the results in several levels, including a map function that outputs an array as the key, outputs another value (1 in this case) as `value` (**Listing 8**), and queries the appropriate grouping parameters `group_level=<x>`. A `group_level` of `0` does not group at all, and the reduce function in the example only returns a *4*. If you set `group_level=1`, the first entry in the array is used for grouping and counting, whereas `group_level=2` tells CouchDB to combine the first two entries and count them in groups.

Of course, this can be combined with the familiar `startkey` and `endkey` (e.g., to evaluate only the `gardenrobot-1` source). Again note the logic that the end key is not included in the output. You need an end key that is greater than `gardenrobot-1`. Because you do not know what the next largest key is, the end key must be either `gardenrobot-1x` or `gardenrobot-1,{}`, because an empty object ranks higher than any string.

Replication

The *messages* database is the central point where all status messages are received. However, for performance or storage reasons, it makes sense to store the sorted status messages in different databases: one database for IoT, one for weather warnings, and another for server messages. Thanks to CouchDB's replication capabilities, a solution is easily found. To begin, create three additional databases as described at the beginning of the article: *iot_messages*, *weather_messages*, and *server_messages*.

Each database in a CouchDB installation can act both as a replication server and as a replication client.

Two databases can also replicate each other. During replication, it does not matter whether or not the databases are in the same CouchDB instance.

You can even set up a replication of a database from instance 2 to instance 3 in CouchDB instance 1.

Replication takes place from the changes feed of a CouchDB database, which is where the document IDs of changed (or newly created) documents are stored. Creating, updating, or deleting a document appends the ID and revision number of the document to the changes feed. Previous entries for a document disappear from the feed so that each document ID only appears once.

For replications from Source to Target, the revision number of a changed source document must be greater than the revision number of a target document for the replication to be executed. A small example will illustrate

Listing 8: Reduce and Group

```
$ curl -X GET localhost:5984/messages/_design/queries/_view/by-source-type-count?group_level=1
{"rows":[
{"key":["18739949083333"],"value":1},
{"key":["gardenrobot-1"],"value":2},
{"key":["server"],"value":1}
]}
```

this important point: Start a one-time replication from the database Source to the database Target (but not back). The Source database receives the new document,

```
{"_id":"d1","_rev":"1-1k9xyc",↩
 "name":"Kurowski"}
```

which is now replicated in the Target database, where the identical document is created. A change then occurs in the Target database, which results in a revision number increase in Target:

```
{"_id":"d1","_rev":"2-7ks1121",↩
 "name":"Oliver Kurowski"}
```

If the document is also modified in Source, no replication to Target follows, because the revision number is not higher, but the same. In this case, the data is not consistent: Two different documents have the same ID and version number. However, when the file in Source is changed a second time, the revision number increases to `3-…`, and the document can be replicated to Target.

Deletions of documents are not actually deletions, either. Instead, the database tags the document as `_deleted:true`, and the revision number is incremented and can then no longer be called up. If the document is deleted on Target in the previous case, it is given a higher revision number, like a change, and replicating it again would be unsuccessful. You would need to modify the document in Source again for the revision number to continue to increment so that the document can again be replicated.

Conclusions

This article has probably only whet your appetite for opportunities with CouchDB. MapReduce is a powerful paradigm for designing queries against large volumes of data, and it is particularly powerful when combined with grouping.

There are many other interesting topics related to CouchDB that I have not covered. In addition to attachments is the ability to attach binary data (e.g., images and other formats) to, and retrieve it from, a document. Also, Mango is the declarative CouchDB query language that allows data to be queried without MapReduce. Finally, the cluster and replication model with configurable shards and copies, among other things, is a strong argument for getting to know the database. I hope you have fun doing so! ■

Info

[1] Installation: [https://couchdb.apache.org]

[2] MapReduce: [https://de.slideshare.net/slideshow/couchdb-mapreduce-13321353/13321353]

KubeVirt integration in OpenShift and Rancher

Best of Both Worlds

We describe how OpenShift and Rancher use their different architectures to integrate KubeVirt, an extension used by Kubernetes to operate virtual machines in addition to containers. By Andreas Stolzenberger

Traditional enterprise virtualization clusters run on physical servers, on which Kubernetes clusters, whose nodes run on virtual machines (VMs), are then based. This kind of architecture is fraught with many drawbacks. On the one hand, two layers means additional license and subscription costs. On the other hand, both the VM and container layers virtualize resources such as networks and storage, which impairs performance in many installations.

In some organizations, Kubernetes nodes already make up the majority of VMs in the cluster while the number of classic VM-based applications steadily decrease. The tide has turned from many VMs and a few containers to many containers and a few VMs. Kubernetes can take care of these remaining VMs with an add-on named KubeVirt, which allows you to remove the obsolete virtualization layer from the data center and place the server hardware with all the containers and VMs entirely under Kubernetes' control.

Add-Ons for VMs

Although you usually think of containers when you hear the word Kubernetes, strictly speaking, Kubernetes is an open cluster management framework. Given the right choice of custom resource definitions (CRDs) and add-ons, the framework is not only capable of managing containers but can also handle virtual networks, distributed storage, and, of course, the matching security rules and access authorizations.

Extensions for Kubernetes save their current configurations in the system's own key-value store (typically `etcd`) and provide a REST API. They use these configurations to field instructions from the Kubernetes framework. To use the add-on features, you need to formulate a matching request to the Kubernetes framework. The "child" used in the request (i.e., the resource type to be managed) then points to the add-on API to which Kubernetes transfers the request.

Extensions also use the Kubernetes framework to communicate with each other, which makes it easier to abstract the resources. Where an add-on needs access to a persistent volume, for example, it makes this request to the storage class managed by Kubernetes. The requesting add-on does not need to know which storage driver is running in the cluster or how it provides the volume. Kubernetes also abstracts network resources according to the same principle.

KubeVirt Basics

KubeVirt [1] is integrated into the framework like all other add-ons. Red Hat founded the open source project back in 2016 as a follow-up project to the oVirt enterprise virtualization system, where development work was discontinued at the end of 2022. On the back end, KubeVirt uses well-known and established technologies such as libvirt and the Qemu and KVM stack. KVM controls the virtualization functions of the CPUs, while Qemu simulates the machine and its peripherals. As a result, KubeVirt can run on any system whose CPU provides virtualization functionality. You can also use a VM for a trial installation, provided it is capable of nested virtualization (i.e., running a VM in a VM).

To avoid the need for KubeVirt to change the host operating system setup, it bundles the Qemu KVM stack's virtualization functions themselves into containers. These containers do need to be running in privileged pods on the respective host to be able access the host CPUs' virtualization features. On the client side, KubeVirt introduces another command-line interface (CLI) tool, `virtctl`, for controlling the VMs in addition to the regular `kubectl` command.

Photo by Darren Halstead on Unsplash

As a fully integrated Kubernetes extension, KubeVirt also offers enterprise-level VM features such as automatic VM failover if a host goes down. The matching storage add-on, which must provide persistent volumes (PVs) as block storage for KubeVirt, takes care of data carrier redundancy. A simple filesystem such as NFS does not work here.

If you are familiar with Kubernetes, you can set up KubeVirt manually. All you need is a K3s or MicroShift single-node setup, a little patience, and many lines of YAML definitions. Although a manual installation will work, it does not lend itself to ease of use. KubeVirt works better when integrated into a distribution. The two major Kubernetes distributions, OpenShift (Red Hat) and Rancher (SUSE), use very different architectures to integrate KubeVirt.

KubeVirt on OpenShift

In Red Hat OpenShift, or its free variant OKD, KubeVirt can be installed easily on an existing cluster from the OperatorHub registry platform – assuming the cluster is running on physical servers or the OpenShift nodes support nested virtualization. As mentioned earlier, KubeVirt requires a storage class that can provide block PVs for VMs.

To build VMs, you will first want to create appropriate templates. The KubeVirt operator comes with a catalog of templates, and it can clone existing RAW or QCOW disk images (Figure 1). Alternatively, you can install the VM in the traditional way (e.g., from an operating system ISO).

Qemu simulates the same hardware and firmware on the KubeVirt VM as is also used on conventional KVM desktop VMs. The Q35 machine type can also run the latest Windows systems thanks to UEFI secure boot. Paravirtualized VirtIO devices take care of storage and LAN adapters on the VM.

Node tags let you decide which nodes in the OpenShift or OKD cluster can run VMs, which means that organizations can decide whether they want to run container and VM workloads together on a single node or prefer to separate VMs and application pods and assign them to different node groups. OpenShift and OKD can use all supported and block-storage-capable drivers as storage. In hyperconverged scenarios, most admins will likely use the OpenShift storage operator, which is based on the Rook and Ceph open source projects. However, external tools such as Trident for NetApp storage systems also work.

The HyperShift project is also of interest for service providers. It uses the KubeVirt operator's VM functions to run several separate virtualized OpenShift clusters on a single physical cluster. In this way, several clusters can be operated with separate root access, yet still share underlying resources such as Ceph storage.

Integration with Red Hat

The integration of KubeVirt on OpenShift leaves little to be desired. The web graphical user interface (GUI) offers many convenient functions, even with some compromises compared with traditional enterprise virtualization platforms. For example, if you want to make changes to a VM's disk assignments once the definition is completed, you won't find all the functions you need in the graphical interface. If worst comes to worst, you will find yourself back editing the YAML code in legacy Kubernetes style.

Additionally, you might find that a soft VM shutdown in the web GUI fails to work. The user interface does not have a *Force Power Down* button. In cases like this, the workaround is the CLI, where you can force the recalcitrant VM to shut down by typing:

```
virtctl stop <VM-Name> --force
```

Generally speaking, you need to remember that combining OpenShift, OpenShift virtualization, and OpenShift storage creates a large and resource-intensive setup of the type that is more likely to be found in large data center installations. A stack like that is not a good choice

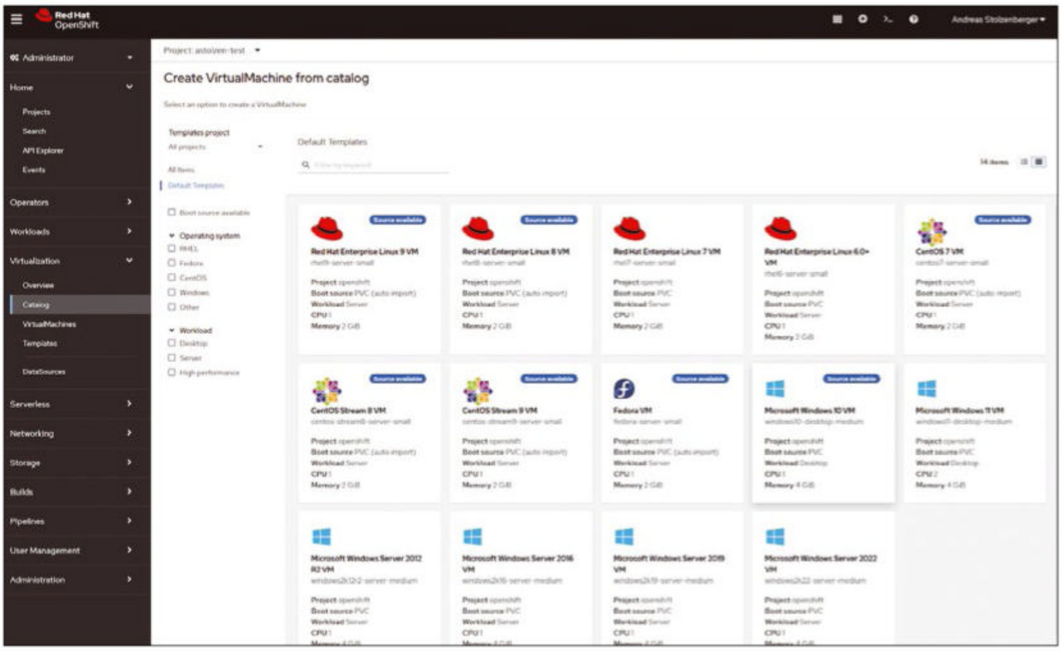


Figure 1: OpenShift and OKD seamlessly integrate KubeVirt's virtualization functions into the GUI, providing a range of ready-made VM templates.

for a small three- to five-node environment in a branch office.

Rancher Harvester

SUSE implements KubeVirt a little differently by relying on the separate hyperconverged tool Harvester. Much like the CoreOS project (Fedora/RHEL), SUSE uses a lean immutable Linux distribution named Elemental for SLES and openSUSE Leap. Elemental only includes the supported hardware drivers, the kernel, a container runtime, and Kubernetes. All other functions are containerized. If you are interested in building your own immutable images, note that SUSE offers the Elemental toolkit [2] for this purpose. The images can then be used to establish lean K3s cluster nodes, for example.

The Harvester distribution combines an Elemental image with the basic Kubernetes functions, Longhorn (distributed iSCSI storage for Kubernetes), KubeVirt, and a simple web UI similar to that in Rancher. During the straightforward Harvester installation from the Elemental boot image, you can choose whether to create a new cluster or add the node to an existing cluster. The first node is the web GUI on a virtual IP address, which means that other nodes in the cluster can also take over this role. For simple test setups, though, Harvester will already work with a single node.

Once installed, Harvester comes up with a very simple interface that is reminiscent of familiar enterprise virtualization environments. Users can upload ISO images and create and start new VMs in just a few steps.

The VM GUIs are accessed from a browser-based virtual network computing (VNC) console. The usual functions (e.g., snapshots and live migration) are also available. You probably will not even notice the Kubernetes underpinnings. Also, you have no option in Harvester to run regular container workloads.

The environment uses a LAN adapter on the Harvester nodes for its management network. You can then assign further network interfaces to bridged VM networks. VMs on these virtual LANs then have direct network access.

Much like OpenShift, the Harvester GUI occasionally lacks functionality. For example, if you want to change the boot sequence of the data carriers after installing a VM, you will not find an option for doing so; instead, you have to edit the YAML code.

VMs? Yes; Containers? No

With the feature set it offers, Harvester is a standalone hyperconverged approach for VMs only, but a Harvester cluster can be integrated into an existing Rancher environment with just a few steps. If your organization already uses Rancher, you can set up the connection to Harvester and then use the Rancher GUI to manage the Harvester functions. Things only start to get really interesting if you use the Harvester node driver. Rancher relies on this driver to steer the process of rolling out and managing additional Kubernetes clusters with K3s or RKE2 nodes running on Harvester.

You first need to configure the desired cluster configuration in a menu – that is, define the number of nodes you will be using, the Linux and Kubernetes distributions the cluster will use, and the resources (e.g., disks, CPUs, and memory) you want to assign to the VMs. Rancher then rolls out all the required VMs on Harvester, configures the environment, and adopts the newly rolled out Kubernetes clusters into a Rancher multicluster management setup. These virtual Kubernetes clusters on Harvester VMs can share resources such as underlying Longhorn storage.

As mentioned, Kubernetes initially only provides a hyperconverged cluster for VMs and distributed storage under the Harvester hood. Application containers are not used. The open source tool offers a solid feature set with comparatively frugal hardware requirements, which means it can easily serve as a drop-in replacement for expensive vSphere installations. Harvester really starts to buzz in combination with Rancher-flavored multicloud management, allowing you to automate the process of rolling out and managing multiple Kubernetes clusters on Harvester.

Conclusions

Although OpenShift and Rancher use the same toolkit to solve VM integration with KubeVirt, the architectures are very different. Whereas OpenShift integrates VMs directly and supports parallel operation of VMs and pods on the same nodes, Rancher separates the VM platform from the container platform with Harvester on its own nodes. Both approaches have their advantages and disadvantages. What I like about Harvester is that it is also a great choice as a vSphere replacement in small to medium-sized installations. The Rancher management integration is a good thing to have, especially when it comes to managing multiple virtual Rancher clusters on the Harvester platform.

OpenShift's virtually seamless integration of VMs with containerized applications is a great success. From the administrator's point of view, there is hardly any difference between an application in a container pod or a VM, especially given that containers and VMs can run alongside each other on virtual networks. That said, OpenShift, including the virtualization component, targets larger infrastructures, and only makes sense in environments with more than eight nodes. ■

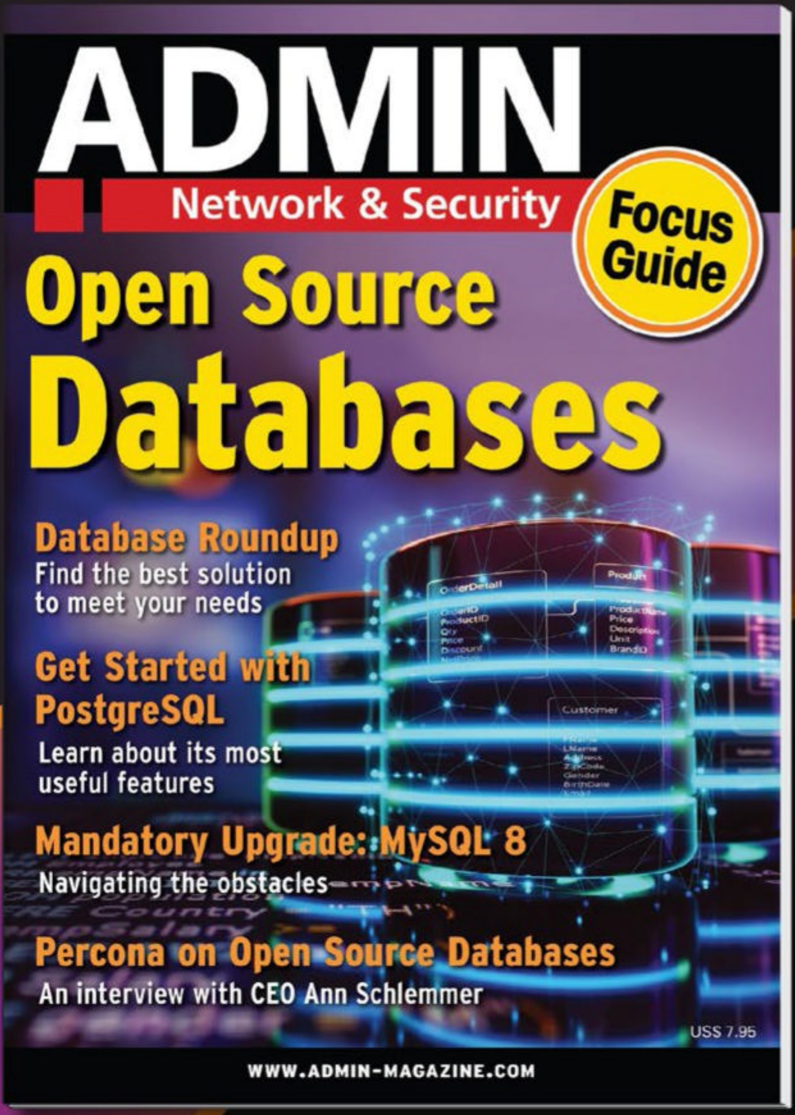
Info

[1] KubeVirt: [https://kubevirt.io]

[2] Elemental toolkit: [https://github.com/rancher/elemental-toolkit]

The Author

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



Production-ready mini-Kubernetes installations

Rich Harvest

Kubernetes can be highly complex, with massive setup routines that are totally over the top for newcomers. If you want to try out Kubernetes or run it in production, you have a number of options, even if you decide not to use the comprehensive packages from established vendors. By Martin Loschwitz

No matter what your technical problems, containers make everything easier, better, and faster – at least that's the rosy promise of the glossy brochures. However, some distributors of container platforms that include Kubernetes provide an excellent tool that, once it has been rolled out, keeps on adding components that can quickly overwhelm newcomers with its complexity to cover various missing aspects of container operation.

Each of these components, whether addressed by the tool itself or as an add-on, comes with its own system requirements and drags in other components that also need to be rolled out before anything else happens. The many solutions in a provider's portfolio did not come about by chance but are part of a carefully developed product strategy. The idea is to cover every eventuality, discourage admins from shopping around, and ensure maximum returns.

By the time you have implemented a production-ready Kubernetes (K8s) in this way, you will have paid a large amount of cash to the distributor and your choice of hardware vendor, fought your way through countless pages of documentation, and probably worked for weeks to knock all the add-ons and the tool itself into a production-ready state. It's easy to understand why many admins lose interest in containerization before they have even gotten started.

In this article, I focus on Kubernetes per se and the means for achieving the smallest possible K8s setup that is as complete as possible for a production or development environment.

Basics

Any container platform certainly needs scalable, redundant storage; versatile software-defined networking; and comprehensive security features. The obvious question is: How do you set up Kubernetes without breaking the bank? What options do you have for building an executable Kubernetes that is suitable for production use outside the sphere of influence of the major distributions? How do you create a small but capable K8s playground for your first steps?

Plenty of ready-made solutions can address all of these challenges; after all, K8s business has been a revenue driver for countless technology groups for years. However, keeping track and developing a suitable strategy can be a challenge.

Test or Production?

Even if you only want to build a small Kubernetes for your own needs, you must ask a central question: Does the setup need to be suitable for production use? Much depends on the answer.

Lead Image © Goran Bogicevic, Fotolia.com

Intuitively, the vast majority of administrators will probably think of redundancy first, and Kubernetes is all about redundancy on several levels. The very principle of a distributed and scalable platform is diametrically opposed to a solution without any redundancy. First is the redundancy that Kubernetes itself requires. Central components such as the scheduler, the Kubernetes API, and the cluster manager need to be operated redundantly in production setups; otherwise, the failure of individual systems will wreak havoc across the entire platform.

Second, a redundant Kubernetes is useless if storage for the running container instances is not set up to be redundant; again, its failure would paralyze the entire platform. Whether redundant or not, Kubernetes can be rolled out by almost identical methods, either as a single-node setup for development purposes or as a highly redundant installation. Obviously a redundant setup requires more hardware than a simple development environment, but in both cases the installation can at least be virtual.

The infrastructure surrounding a Kubernetes installation requires more thought. In the context of a development environment, the most important consideration is that you need a way to create standards-compliant volumes in Kubernetes (persistent volumes and persistent volume claims). Whether they then point to a local logical volume manager (LVM) volume in the background is pretty much irrelevant.

However, this situation changes if you are looking at a production setup. Redundant storage can be implemented in various ways. Besides legacy network-attached storage (NAS) or a storage area network (SAN), you can use more modern approaches such as Ceph or Longhorn. However, you do need to procure, install, and set up the required hardware before you can get started with Kubernetes itself.

In the example here, I assume that at least five servers are available for the production scenario: three systems for the Kubernetes control plane, a Ceph-based distributed storage system, and two additional systems for (redundant) container operations.

Beware of Vanilla

A warning is appropriate at this point: Administrators who have not yet had any experience in the K8s world, in particular, tend to stumble into a trap by rolling out the vanilla Kubernetes distribution (i.e., the containerized software that can be found on the Kubernetes homepage [1]). As things stand at present, though, this approach is strongly discouraged – an opinion shared in the Kubernetes community.

Since its inception, K8s has been based on the principle that an official Kubernetes version exists whose central role is to specify the Kubernetes API and define the rules. Although the open source upstream (vanilla) Kubernetes can be rolled out, you won't enjoy such a setup for long, because it explicitly gives you only the absolute minimum of services to operate with minimum redundancy. The management tools are missing, as are tools for connecting the basic Kubernetes installation with external services such as software-defined storage (SDS) or software-defined networking (SDN).

Although technically possible, a great deal of effort is required to create an environment that is production- or even just development-ready. Experienced Kubernetes admins go so far as to describe running a local vanilla Kubernetes as a form of administrative self-boycott. If the experts already fear such a task, K8s newcomers are very much advised not to tread this path. You do not need to take such a course of action: Far removed from the huge Kubernetes environments are rugged and agile solutions that enable the operation of a standards-compatible K8s cluster.

Many Options

Development setups kick off this tour of the mini-K8s universe. In these setups, you typically want to create a local Kubernetes instance to test your workloads and check whether your configuration works in the intended way.

The Kubernetes market can be confusing here, with various solutions vying for your favor, most of which claim to be just as suitable for small development environments as for redundant production setups in data centers. The real problem is a lack of clarity: If you haven't had much experience with Kubernetes in the past and just want to familiarize yourself with the principles, you can quickly become confused when you hear about K3s [2], k0s [3], minikube [4], Microkubes [5], and the many other variants.

These names relate to K8s distributions, all of which enrich the vanilla version of Kubernetes with their own components and setup logic and set up the whole kit and caboodle so you can get places with just a little typing at the command line. Where they differ – if at all – is mainly in the components they use to enrich the environment. As crazy as it sounds: "Not invented here" plays a major roll in the colorful world of Kubernetes. Vendors often only decide to launch their own K8s distributions because they want to create a unique selling point compared with other manufacturers, which, from the point of view of the K8s newcomer, doesn't make things any easier.

If you want to start with a local setup, you first need to set up a classic virtual machine (VM). Ideally, you will want to run Debian GNU/Linux 12 or Ubuntu 24.04. Preferably, you should also install a runtime environment on the system that lets you operate containers on Linux. In both cases, the choice will typically be the Docker Community Edition (Docker CE). More modern distributions such as Ubuntu 24.04 come with a runtime environment in place, in the form of Podman, which can be used immediately.

One thing that practically all mini K8s distributions have in common is that their authors attach great importance to getting a complete Kubernetes up

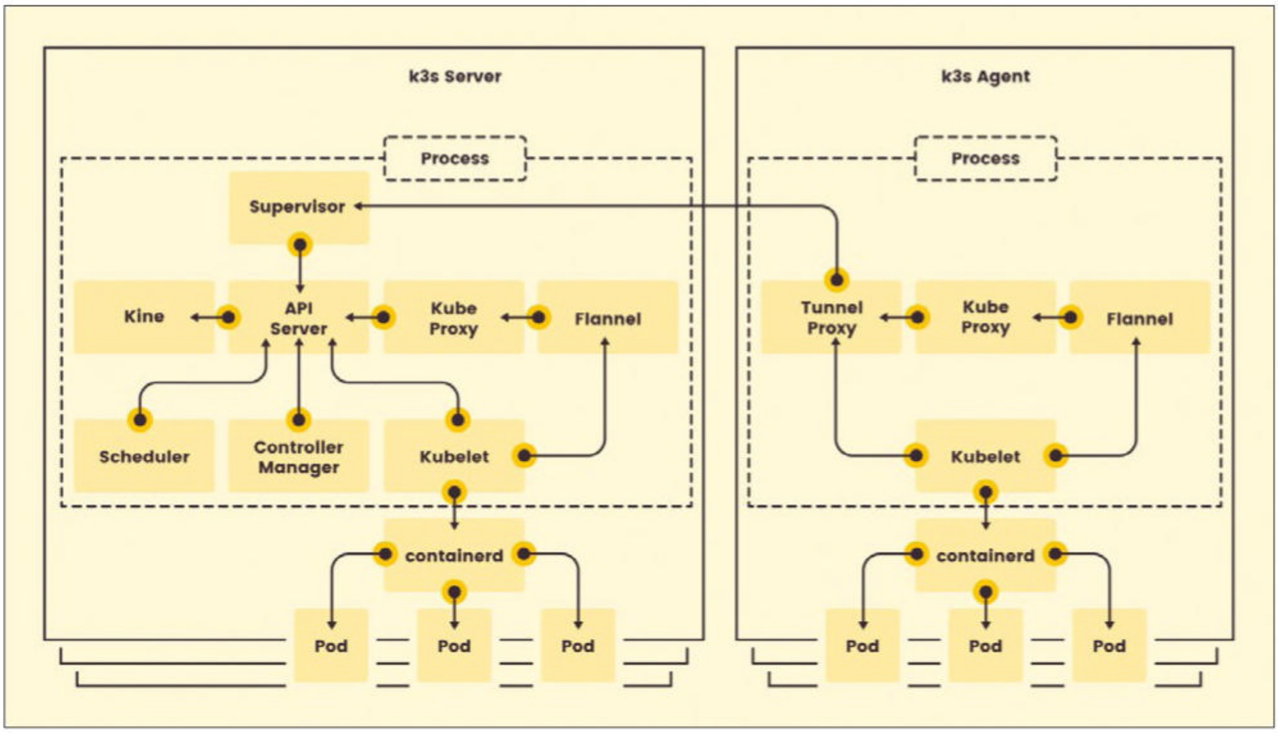


Figure 1: K3s lets you roll out Kubernetes in small environments for development or production purposes. © K3s

and running as quickly as possible.

It comes as little surprise to hear that K3s, for example, can be set up on an existing virtual instance in next to no time (Figure 1). K3s is a significantly pared down distribution, originally created by the Rancher developers, that is still fully compatible with the Kubernetes API. In the meantime, the focus has shifted to edge computing, IoT, and continuous integration and continuous delivery (CI/CD) applications.

Installing K3s is very easy (see the "Insecure Installation" box): A short `curl` command is all it takes to roll out all the components in a single-node installation and, after about 30 seconds, view a list of the available K3s nodes on the screen:

```
$ curl -sfL https://get.k3s.io | sh --
$ sudo k3s kubectl get node
```

In the standard installation, though, this simply means the Kubernetes control plane itself.

Right now, you are still missing a K3s agent that can run containers. To do this, first find the token for authenticating agents and then launch the virtual instance, specifying its address:

```
$ NODE_TOKEN=↩
  $(cat /var/lib/rancher/k3s/server/token)
$ sudo k3s agent ↩
  --server https://<Server-IP>:6443 ↩
  --token ${NODE_TOKEN}
```

These commands produce a complete Kubernetes that can create volumes and manage containers on the specified IP. If your VM is powerful enough, you can start experimenting right now with something quite close to a production setup.

At first glance, you will probably not realize how many components K3s integrates out of the box. I already mentioned SDS and SDN; K3s supplies working components for both: Volumes can be created, locally at least, by an LVM provider, and Flannel software gives you SDN. Although it is considered one of the simpler SDN solutions for Kubernetes, it is fine for test setups and for the vast majority of production environments. Flannel can also be combined with various network components within Kubernetes (e.g., with the Istio mesh solution).

By the way, you are not restricted to rolling out K3s in single-node mode. If you later want to recreate a development K3s setup in production, you can call the setup script with various parameters that roll out the Kubernetes control plane redundantly. You can tell that K3s was originally the mini-K8s distribution in the background with Rancher and was designed for this purpose. I'll get back to Rancher later.

kOs Impresses

The situation with k0s is very similar to that with K3s (Figure 2). This Kubernetes distribution by Mirantis, a cloud computing company, is also available as free software and can be installed in a few simple steps. Here,

Insecure Installation

In terms of installation, both K3s and k0s deserve an admonishment. The principle of loading a shell script from the network with `curl` and using `sudo` to execute it directly as root on the target host naturally causes any security-conscious admin to break out in a sweat; you would want to download the script manually and examine it meticulously before running it. However, because you are using `curl` to set up Kubernetes on a virtual instance, the damage would not be so great in case of an incident and could be simply remedied by ditching the instance and creating a new one.

too, it is advisable to start a single virtual instance for development and test purposes on which to carry out the work. Again, Ubuntu 22.04 is a good choice. The steps are quickly completed: Use `curl` to download the k0s binary, which you then use to roll out the Kubernetes cluster:

```
$ curl -sSLf https://get.k0s.sh | sudo sh
```

If k0s is available locally, install a single-node control plane for Kubernetes and then start the desired services and check their status:

```
$ sudo k0s install controller --single
$ sudo k0s start
$ sudo k0s status
```

The `kubectl` command-line tool can be deployed at this point.

Like K3s, k0s also claims to be suitable for production environments. By running the command

```
$ k0s init > k0sctl.yaml
```

after the install, k0s creates a basic configuration for a cluster consisting of two nodes in the file `k0sctl.yaml`. To make the control plane highly available, and therefore suitable for production, you can edit the file accordingly and then implement the saved configuration, output a configuration file compatible with `kubectl`, and display the running pods:

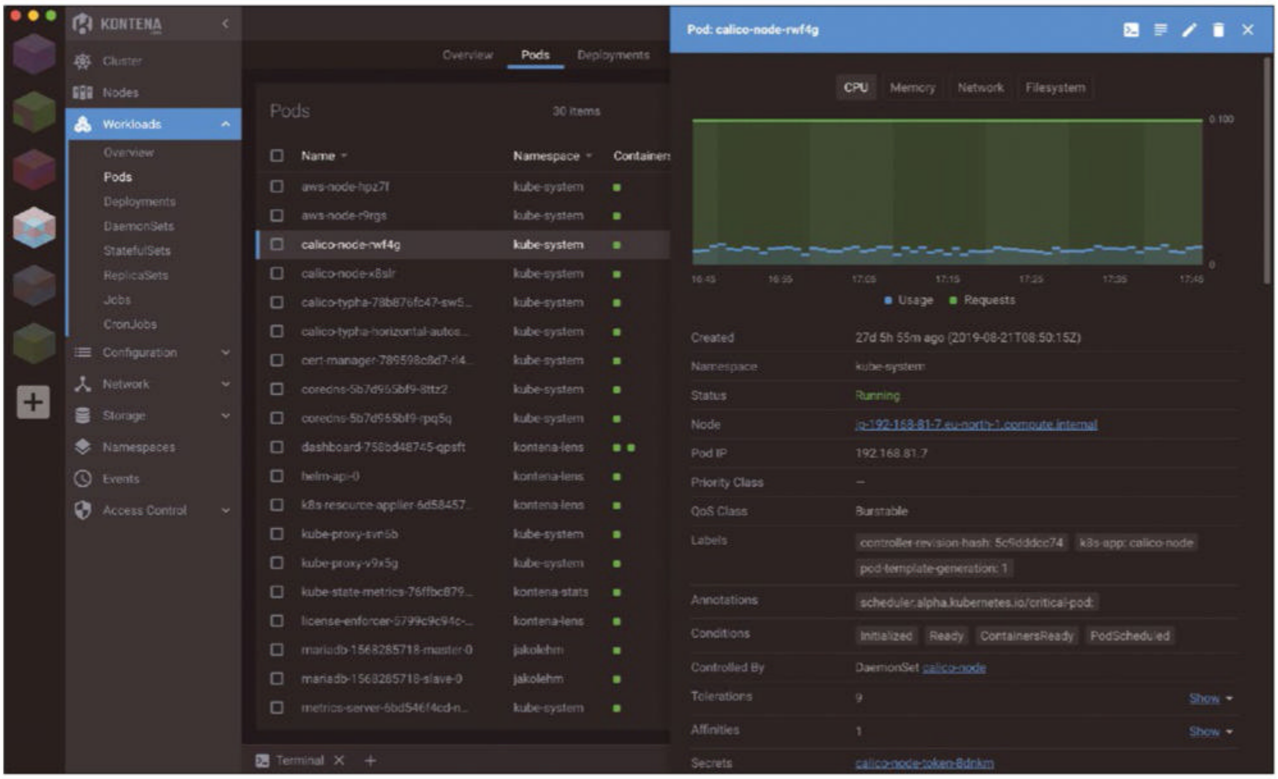
```
$ k0sctl apply --config k0sctl.yaml
$ k0sctl kubeconfig > kubeconfig
$ kubectl get pods ↵
  --kubeconfig kubeconfig -A
```

Like K3s, k0s comes with ready-made integrations for SDN and SDS. However, it takes a different approach, at least in terms of the network, by giving you Calico, which is far more comprehensive and complex than Flannel. When it comes to storage, you face a little DIY. The k0s developers recommend combining the tool with OpenEBS. However, this block-mode storage platform was dropped from the k0s scope of delivery a while back and, according to the authors, needs to be installed by the Helm package manager. The k0s documentation [6] contains instructions for this task.

My choice of describing K3s and k0s for this article has nothing to do with bias. In fact, the various Kubernetes distributions in the same league are not too different in terms of content and technology. If you choose minikube or MicroK8s instead of K3s or k0s, you will quickly realize that their installation procedures are similar. After completing the install, you will have a comparable feature set in each case, and the way things are handled is pretty much the same.

Production, Fast

If you are looking for a flexible and not overly complex solution to roll out a production Kubernetes setup for your company, you first need to make sure the conditions mentioned earlier for a production Kubernetes are met. You need to be clear about the SDS solution you will be using and about the SDN solution you intend to deploy. For the sake of simplicity, assume in this example that the all-rounder Flannel will be handling SDN, with Ceph shouldering the SDS load. Ceph can be rolled out as a service in



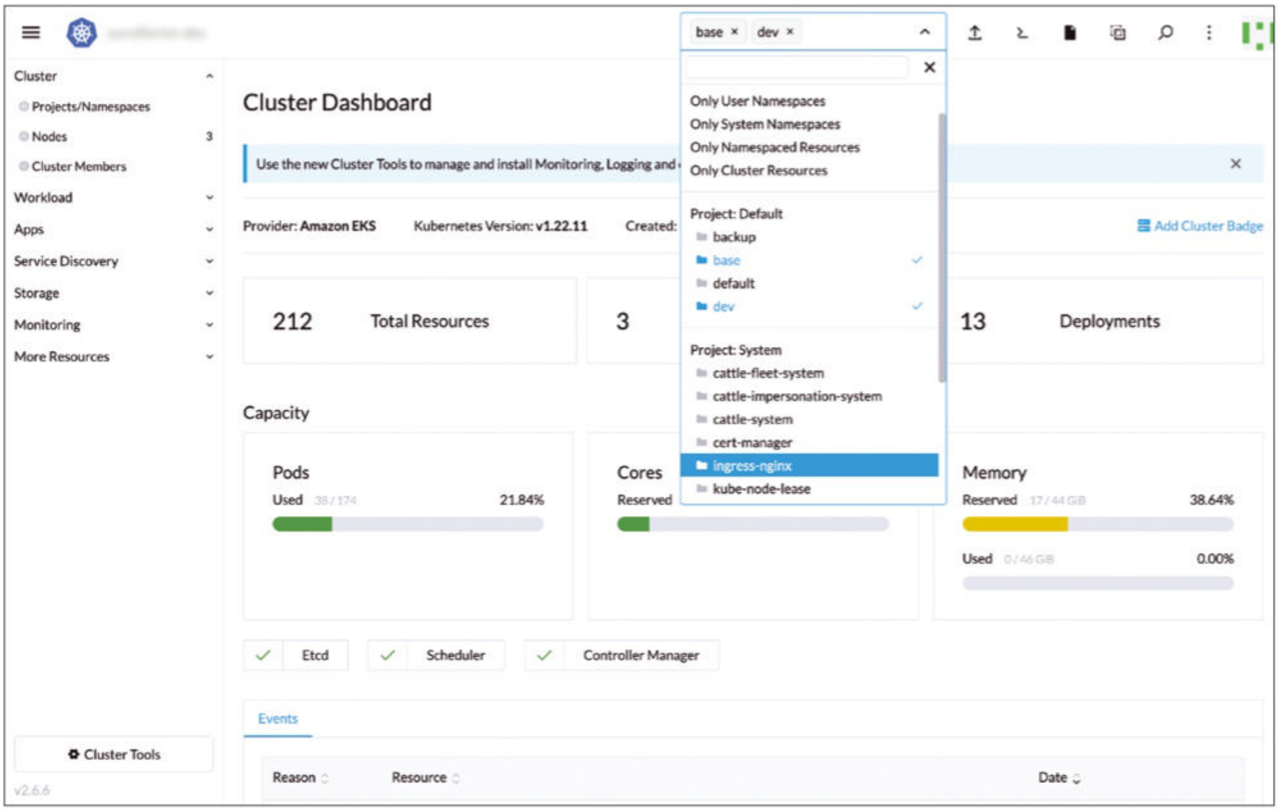
Pods 30 items

Name	Namespace	Containers
aws-node-hpz7f	kube-system	●
aws-node-r9rgs	kube-system	●
calico-node-rwf4g	kube-system	●
calico-node-x8slr	kube-system	●
calico-typha-78b876fc47-sw5...	kube-system	●
calico-typha-horizontal-autos...	kube-system	●
cert-manager-789598c8d7-rl4...	kube-system	●
coredns-5b7d955bf9-8ttz2	kube-system	●
coredns-5b7d955bf9-pq5q	kube-system	●
dashboard-758bd48745-qpsft	kontena-lens	● ●
helm-api-0	kontena-lens	●
k8s-resource-applier-6d58457...	kontena-lens	●
kube-proxy-svn5b	kube-system	●
kube-proxy-v9x5g	kube-system	●
kube-state-metrics-76ffbc879...	kontena-stats	●
license-enforcer-5799c9c94c-...	kontena-lens	●
mariadb-1568285718-master-0	jakolehm	●
mariadb-1568285718-slave-0	jakolehm	●
metrics-server-6bd546f4cd-n...	kube-system	●

Pod: calico-node-rwf4g

Created	27d 5h 55m ago (2019-08-21T08:50:15Z)
Namespace	kube-system
Status	Running
Node	ip-192-168-81-7.eu-north-1.compute.internal
Pod IP	192.168.81.7
Priority Class	–
QoS Class	Burstable
Labels	controller-revision-hash: 5c5dddcc74 k8s-app: calico-node pod-template-generation: 1
Annotations	scheduler.alpha.kubernetes.io/critical-pod:
Conditions	Initialized Ready ContainersReady PodScheduled
Controlled By	DaemonSet calico-node
Tolerations	9
Affinities	1
Secrets	calico-node-token-8dnkm

Figure 2: k0s offers services similar to K3s and enables the operation of a Kubernetes with all the required services in a small footprint. High-availability options are also available for production environments. © Mirantis



Use the new Cluster Tools to manage and install Monitoring, Logging and

Provider: Amazon EKS Kubernetes Version: v1.22.11 Created:

Figure 3: Rancher can roll out fully functional, yet manageable, K8s instances in this useful dashboard. © Magnolia

Kubernetes by Rook [7], which automatically gives you a management option. The SDS part of the setup only plays a minor role in the configuration of Kubernetes itself.

I already looked at Rancher in the context of K3s, but Rancher is also ideal for rolling out production Kubernetes setups (Figure 3). This concept is easy to understand when you think about what Rancher actually is. Rancher is itself based on Kubernetes and integrates a Kubernetes control plane. It can also be used to roll out and control many different Kubernetes instances in parallel. Once you have set up the Rancher control plane, you can add further hosts to the software and then roll out your own small Kubernetes clusters on them with the command-line

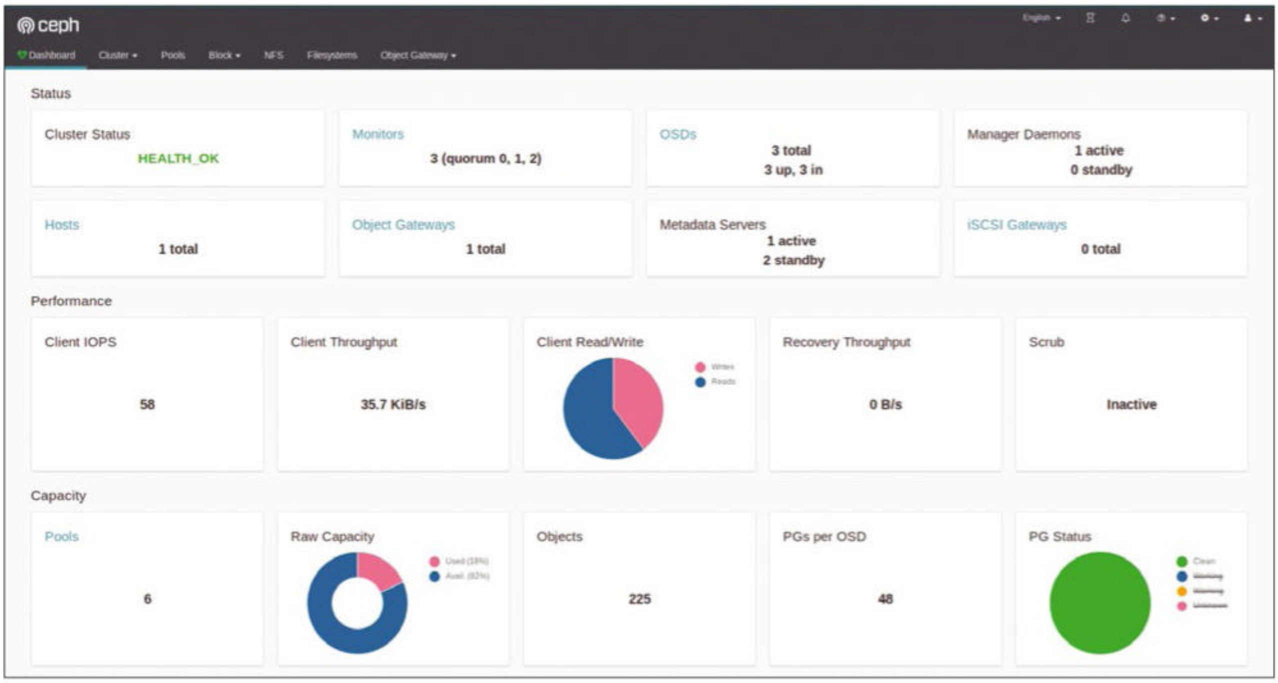


Figure 4: For all K8s setups, Rook is recommended as a storage solution that can be set up in very little time to run in K8s itself. © Rook

interface or graphical user interface. These clusters can then be deleted or reconfigured easily.

What impresses most is that you can set up Rancher just as quickly as K3s or k0s. In practical terms, you simply start a Docker container,

```
$ sudo docker run ⮐
  --privileged -d ⮐
  --restart=unless-stopped ⮐
  -p 80:80 ⮐
  -p 443:443 rancher/rancher
```

and the Rancher web user interface is then available on port 443 of that container. Additional bare metal hosts can now be created here, for which Rancher can use SSH to open connections so that you can make the required configuration changes. It makes sense to prepare all the systems involved at the outset to support a secure shell connection from the Rancher control host.

The operating system install can also be automated by life-cycle management, if so desired. Once you have installed the first high-availability Kubernetes cluster on the target systems with Rancher, you can roll out Rook (Figure 4) for high-availability storage directly afterward.

However, be careful. As I mentioned earlier, the mini-setup described assumes that the controller and storage services are hyperconverged and running on the same systems. Systems with the Kubernetes services of a K8s cluster rolled out by Rancher therefore also need storage devices (hard disks or flash drives) for Rook to offer redundant storage.

Of course, such a setup can be operated on virtual instances – or at least tested there – if you will be using bare metal later in production. Either way, you can achieve a production-ready setup quickly on the basis of Kubernetes with Rancher and Rook, without having to deal with countless add-ons and product variants.

Conclusions

In both development and production operations, Kubernetes can be used to create clusters quickly that comply with all technical standards and that work well.

At this point, however, a word of warning is in order: The work required in the K8s universe includes monitoring workloads, both those of K8s itself and the services running on it. Observability is massively important. Providers know this, and their products install a variety of additional tools (e.g., Prometheus, OpenTelemetry) in addition to Kubernetes for precisely this reason. Security add-ons also play a role.

You will need this functionality even if you opt for one of the less complex solutions presented here, but don't expect to get it straight out of the box. Some follow-up can be expected, at least for production environments.

Given that Kubernetes can be highly complex, you will certainly not want to find yourself flying blind in your own environment.

Regardless, if you want to try out K8s or run it in production, you have a number of options, even if you decide not to use the comprehensive packages from established vendors. ■

Info

[1] Vanilla Kubernetes: [https://kubernetes.io/releases]

[2] K3s: [https://K3s.io]

[3] k0s: [https://k0sproject.io]

[4] minikube: [https://minikube.sigs.K8s.io/docs]

[5] Microkubes: [https://www.microkubes.com]

[6] k0s installation instructions: [https://docs.k0sproject.io/v1.23.6+k0s.2/install/]

[7] Rook documentation: [https://rook.io/docs/rook/latest-release/Getting-Started/quickstart]

The Author

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

Optimizing domain controller security

Leakproof

Configure your domain controller security settings correctly with Policy Analyzer and current Microsoft baselines for a leak-tight Active Directory. By Thomas Joos

Domain controllers (DCs) are a central element of the network architecture; they manage the authentication and authorization of user identities and computers in a Windows domain. Attacks on DCs can be carried out with a variety of methods, including pass-the-hash, exploitation of software vulnerabilities, and insider threats. A compromised DC gives attackers potentially far-reaching access to the network, including the ability to manipulate user accounts, change policies, escalate access authorizations, and steal sensitive data. Moreover, the integrity of the information stored on the network is at risk because attackers are able to manipulate or delete data. Therefore, you need to secure your DCs, as well as the servers and workstations that access them. The free Microsoft Security Compliance Toolkit (SCT) [1] provides an important basis for this endeavor. Active Directory (AD) is one of the most sensitive structures on the network. This central role makes the DC a preferred target for hackers and cybercriminals. Unfortunately, the default domain controller policy responsible for DC security settings only provides rudimentary configurations that often do not offer the protection you need. In this article, I look at how security can be optimized with the help of Microsoft baselines and the free Policy Analyzer.

Securing Networks

The baselines from the SCT are a set of preconfigured security settings based on best practices and expert recommendations. One key benefit of this collection of settings is that it provides a solid foundation for security configuration, significantly reducing the need to research and configure each setting manually, which saves time and resources while making sure the systems are resilient to known threats and attack vectors. The baselines also make it easier to meet legal and industry-specific compliance requirements by suggesting configurations that comply with common security standards and regulations.

Unfortunately, the settings of the Default Domain Controller Policy are pretty rudimentary and do not implement numerous items that Microsoft recommends in its own security baselines. A comparison (which I will come to in a moment) gives you an overview of the important

Photo by Nathan Roser on Unsplash

options that are not set – despite being recommended by Microsoft. You will always want to implement the Microsoft security baselines in all environments. The default settings after installing AD might be guaranteed to work in any environment, but they offer no more than a minimum level of security, and that is just not good enough in these times.

Microsoft offers security settings in the form of SCT baselines that you can distribute automatically by Group Policy, with areas for the Default Domain Controller Policy in AD. This policy specifically protects the domain controller. The toolkit includes the Policy Analyzer, a tool that lets you determine the delta between your current policies and Microsoft's recommendations. It makes sense to compare the settings and, ideally, to implement the specifications to the extent possible.

You can automatically implement the current Microsoft security recommendations as Group Policy templates in the Active Directory for domain controllers, member servers, standalone servers, and workstations. Of course, it is also possible to change individual settings. However, you should only do this if the values are causing issues on your network. In this case, though, it often makes more sense to check why parts of the network or some applications cannot handle the configurations devised by experts. Disabling should be your last resort because it ultimately affects security. In any case, it makes sense to check all the changes in a test environment first, if possible.

Group Policies

Security recommendations are implemented on the basis of group policies that you integrate into Active Directory. To secure servers, simply download the `Windows Server 2022 Security Baseline.zip` file. The Policy Analyzer mentioned earlier is delivered to your computer in the `PolicyAnalyzer.zip` file. The first archive contains various guidelines specific to the Windows server's task, including, for example, special policies for DCs, but also for member servers and for servers that are not part of AD domains.

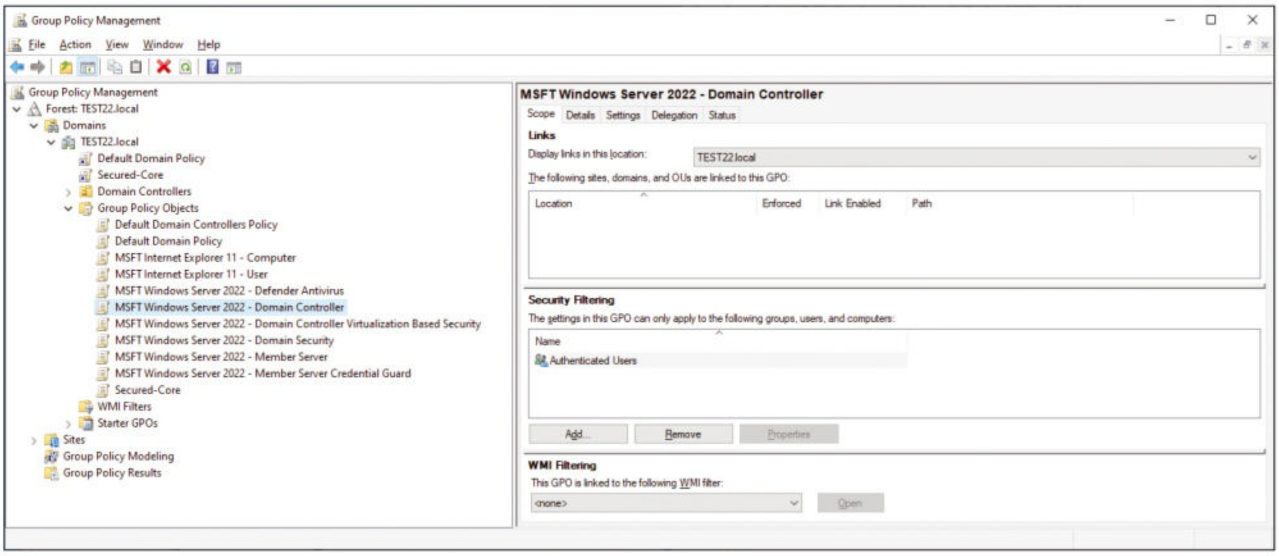
Documentation in the form of Excel tables is included in the scope of delivery. The `New Settings in Windows Server 2022.xlsx` spreadsheet documents all configurations for Windows Server 2022, including the registry items that the policy settings change. All settings are available in `FINAL-MS Security Baseline Windows Server 2022.xlsx`. Several tabs are available for this at the bottom of the spreadsheet. The `GP Reports` directory also contains HTM files for all policies from the SCT that can be opened in your browser. All the configured settings are listed and documented there, giving you a comprehensive picture of what the baselines implement on the network.

Installing Policies on a DC

The use of the policies is simple. SCT comes with a number of ADMX and ADML files to match. You only need to copy these to the `C:\PolicyDefinitions` directory on a DC if they are not already in place, as is the case, for example, if you are implementing policies for Windows 11 23H2, because they are not included in the scope of delivery of Windows Server 2022.

If you work in an organization with central storage locations for ADMX files, you will need to copy the SCT templates to this directory. The ADML files contain the language information of the policy in question. Microsoft provides a `Baseline-ADImport.ps1` script to help you import policies by copying the required files into the correct directories. However, this does not mean that the servers will execute the policies; it only ensures that all the required files are in place in a production environment.

The script also imports the new group policies into AD, but does not link them, which means they are not active as yet but are available for editing. Implementation takes place later when the settings are imported into new or existing group policies (e.g., the *Default Domain Controllers Policy*)


Group Policy Management
File Action View Window Help
Scope Details Settings Delegation Status
Add... Remove Properties
Open

Group Policy Management
- Forest: TEST22.local
  - Domains
    - TEST22.local
      - Default Domain Policy
      - Secured-Core
      - Domain Controllers
      - Group Policy Objects
        - Default Domain Controllers Policy
        - Default Domain Policy
        - MSFT Internet Explorer 11 - Computer
        - MSFT Internet Explorer 11 - User
        - MSFT Windows Server 2022 - Defender Antivirus
        - MSFT Windows Server 2022 - Domain Controller
        - MSFT Windows Server 2022 - Domain Controller Virtualization Based Security
        - MSFT Windows Server 2022 - Domain Security
        - MSFT Windows Server 2022 - Member Server
        - MSFT Windows Server 2022 - Member Server Credential Guard
        - Secured-Core
      - WMI Filters
      - Starter GPOs
  - Sites
- Group Policy Modeling
- Group Policy Results

MSFT Windows Server 2022 - Domain Controller

Links

Display links in this location: TEST22.local

The following sites, domains, and OUs are linked to this GPO:

Location	Enforced	Link Enabled	Path

Security Filtering

The settings in this GPO can only apply to the following groups, users, and computers:

Name
Authenticated Users

WMI Filtering

This GPO is linked to the following WMI filter:

<none>

Figure 1: The Microsoft baselines are implemented as a Group Policy, with the option of flexible configuration.

or by linking the imported policies to containers or organizational units. The *Default Domain Controllers Policy* is linked to the *Domain Controllers* container in which the DC computer accounts reside, where you can, of course, use and extend AD's standard policies.

Microsoft also offers the *MSFT Windows Server 2022 – Domain Controller* policy (Figure 1), which can be linked to the *Domain Controllers* container parallel to the *Default Domain Controllers Policy*. However, at this point, it is a good idea to proceed very carefully before you set the values from the baselines. You need to make sure the policies do not overwrite each other. Although the settings only apply to the DCs, they affect all computers, services, and users who log on to the DCs. Despite this, it still makes sense to implement the baselines as separate policies. If problems occur, you then just need to disable the new policy. If you import all the new settings into the default policy, things are more difficult.

Implementation of Policy Settings

The easiest way to implement the Microsoft recommendations is to open the Group Policy Management Console (GPMC) on the DC and import the policy backups from the Security Compliance Toolkit directory into your production Active Directory as new policies. Alternatively, you can use the PowerShell scripts I mentioned earlier. You will then need to link the policies to the domain or, in the case of DCs, to the *Domain Controller* container (e.g., by drag and drop).

For the implementation, create a new Group Policy named *Windows Server 2022 Domain Controller* in the GPMC *Group Policy Objects* context menu. In the context menu of the new policy, select *Import Settings*, which launches a wizard. In the wizard, specify the `GPOs` (or `GPO`) directory in the SCT archive. After opening the directory, the wizard displays all group policies located in this directory. Select the desired template to tell the wizard to import all the settings into the new Group Policy object (GPO). You can also create multiple policies. If you used the `Baseline-ADImport.ps1` script, the policies already exist in AD but are not yet linked. In other words, you have several ways to implement the settings on the network.

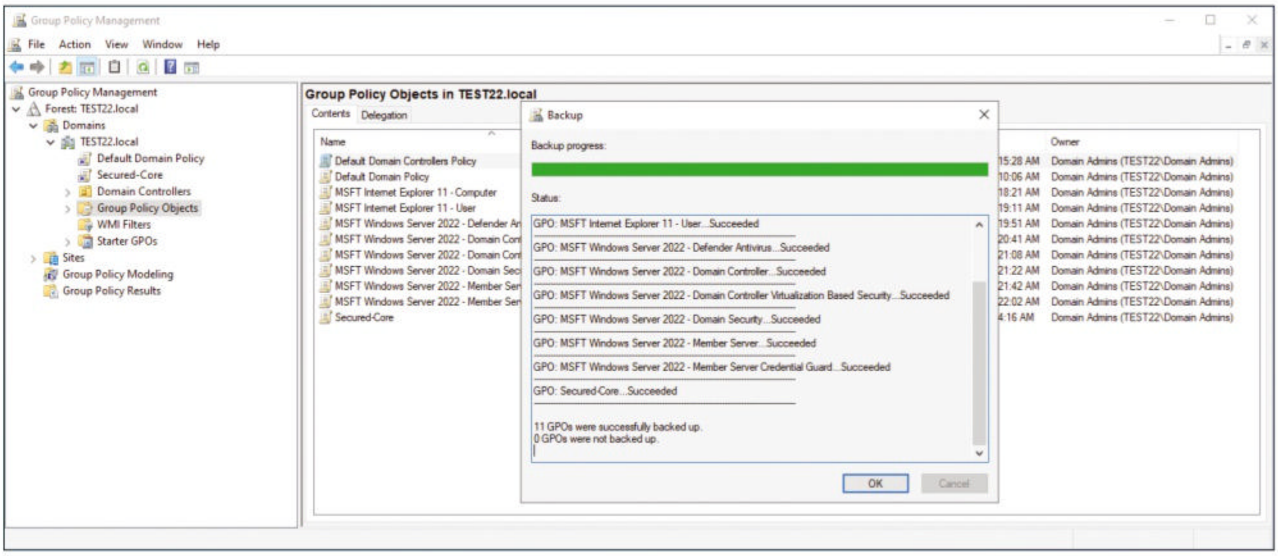
Comparison with Current Configuration

As already mentioned, Policy Analyzer gives you a quick and easy option for comparing different group policies. To do this, you need a backup of the current group policies. To simplify the process, Microsoft also provides policy rules files (`.PolicyRules`). You need `MSFT-WS2022-FINAL.PolicyRules` from the `Documentation` directory of the Windows Server 2022 baseline files. Copy the file to the `Policy Rules` directory, which is located in the `Policy Analyzer` directory.

In a production environment, save your existing policy in the Group Policy Management Console. To back up all policies, simply right-click on *Group Policy Objects* and click *Back Up All*; then, select a directory and save the group policies. As mentioned, this backup serves as the basis for a comparison against the baselines. In this case, the main interest is in the *Default Domain Controllers Policy* (Figure 2). If you like, you can save just this one policy.

Next, launch Policy Analyzer from the download directory. You do not need to install the tool. In Policy Analyzer, get started by clicking on the *Policy Rule sets in* field at the bottom; then, select the `Policy Analyzer` directory and the `Policy Rules` subdirectory. You already copied the policy rules of the current Windows Server 2022 baselines to this directory.

To launch a comparison with your current policy, click *Add*, then select *File | Add files from GPOs*. Navigate to the folder with the backup of your existing group policies and complete the process by selecting *Import*. Then save the process as a `.PolicyRules` file (e.g., as the *Default Domain*


Group Policy Management
File
Action
View
Window
Help
Group Policy Management
Forest: TEST22.local
Domains
TEST22.local
Default Domain Policy
Secured-Core
Domain Controllers
Group Policy Objects
WMI Filters
Starter GPOs
Sites
Group Policy Modeling
Group Policy Results
Group Policy Objects in TEST22.local
Contents
Delegation
Name
Default Domain Controllers Policy
Default Domain Policy
MSFT Internet Explorer 11 - Computer
MSFT Internet Explorer 11 - User
MSFT Windows Server 2022 - Defender An
MSFT Windows Server 2022 - Domain Cont
MSFT Windows Server 2022 - Domain Cont
MSFT Windows Server 2022 - Domain Sec
MSFT Windows Server 2022 - Member Ser
MSFT Windows Server 2022 - Member Ser
Secured-Core
Owner
Backup
Backup progress:
Status:
OK
Cancel

	Owner
15:28 AM	Domain Admins (TEST22\Domain Admins)
10:06 AM	Domain Admins (TEST22\Domain Admins)
18:21 AM	Domain Admins (TEST22\Domain Admins)
19:11 AM	Domain Admins (TEST22\Domain Admins)
19:51 AM	Domain Admins (TEST22\Domain Admins)
20:41 AM	Domain Admins (TEST22\Domain Admins)
21:08 AM	Domain Admins (TEST22\Domain Admins)
21:22 AM	Domain Admins (TEST22\Domain Admins)
21:42 AM	Domain Admins (TEST22\Domain Admins)
22:02 AM	Domain Admins (TEST22\Domain Admins)
4:16 AM	Domain Admins (TEST22\Domain Admins)

```
GPO: MSFT Internet Explorer 11 - User...Succeeded
GPO: MSFT Windows Server 2022 - Defender Antivirus...Succeeded
GPO: MSFT Windows Server 2022 - Domain Controller...Succeeded
GPO: MSFT Windows Server 2022 - Domain Controller Virtualization Based Security...Succeeded
GPO: MSFT Windows Server 2022 - Domain Security...Succeeded
GPO: MSFT Windows Server 2022 - Member Server...Succeeded
GPO: MSFT Windows Server 2022 - Member Server Credential Guard...Succeeded
GPO: Secured-Core...Succeeded

11 GPOs were successfully backed up.
0 GPOs were not backed up.
```

Figure 2: Before comparing the existing rules with those of the baselines, it makes sense to make a backup of the GPOs.

Controllers Policy). This policy is then also visible in Policy Analyzer.

With these steps, you have integrated the baseline policies into Policy Analyzer by their policy rules file and a recent backup of your *Default Domain Controllers Policy*, which you also saved as a policy rules file.

These steps integrate the baseline policies into Policy Analyzer by their policy rules file and the current backup of your *Default Domain Controllers Policy*, which you also saved as a `.PolicyRules` file. For the comparison, select your imported *Default Domain Controllers Policy* followed by *MSFT-WS2022-FINAL*. This is the policy rules file you copied from the current baselines for Windows Server 2022. Do not select any other policies; you only need these two. Now click on the *View/Compare* button. The process takes a few seconds. When done, you will see a precise comparison of your current *Default Domain Controllers Policy* and the baseline.

Adjusting Values for DC Security

The comparison helps you quickly identify strongly recommended options that are not set in your environment. As described earlier, you will generally want to implement the complete policy for your domain controllers. To do this, you can use the policy from the security baselines as a second policy for the DCs. The advantage is that you can leave the default policy unchanged and disable all new settings in one fell swoop if issues occur, simply by disabling the security baseline. The disadvantage is that you have to manage several group policies.

Of course, you could go through the individual differences and transfer them to your default domain controller policy manually, but this is a very time-consuming process and requires painstaking documentation. The most important values can be found under *Computer Configuration* | *Policies* | *Windows Settings* | *Security Settings*. Various folders with security baseline configurations are stored here. Pay particular attention to the settings in *Security Options*, which is where the security baseline governs user accounts and client communication with the domain controller. These settings can also be found in the `MSFT Windows Server 2022 - Domain Controller.htm` file in the `Windows Server 2022 Security Baseline\Windows Server-2022-Security-Baseline-FINAL\GP Reports` directory.

At the same time, the baseline also generates rules for the Windows firewall, which are not included in the default domain controller policy – nor are the numerous logging actions. These rules can be found under *Computer Configuration* | *Policies* | *Windows Settings* | *Security Settings* | *Advanced Audit Policy Configuration* in the group policies. You will generally want to implement these configurations in all environments. Negative effects are unlikely unless the domain controllers are already suffering capacity problems. In this case, extended system monitoring can cause the DCs to respond more slowly, but if you are at this point, replacing the hardware is the order of the day rather than reducing the recommended security levels.

Conclusions

In general, the current Microsoft baselines should always form the basis for securing networks with Windows computers. Caution is always required when implementing policies on domain controllers, or you could experience scenarios in which entire Active Directory environments no longer work correctly. Policy Analyzer is a powerful tool that helps you establish security with group policies and lets you compare several GPOs against each other. You do not have to use all settings, but it makes sense to use as many as possible. ■

Info

[1] Security Compliance Toolkit: [https://www.microsoft.com/en-us/download/details.aspx?id=55319]

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].

Hardening SSH authentication to the max

Keys to the Realm

Public key authentication further supplemented with one-time password or hardware authentication methods improves SSH security while offering genuine convenience. By Thomas Reuß

If you want to open access to SSH, and possibly even to users on the Internet, you need to harden authentication, preferably with a combination of key pairs, discoverable and non-discoverable credentials, multifactor authentication, authenticator apps, and other methods. SSH with public key authentication should be your default setting.

Public Key Authentication

To get started, you need to create an SSH key pair with the `ssh-keygen` command. Even now, the process is fraught with pitfalls: Given that massive attacks on RSA2048 are commonplace, you can expect RSA3072 to become the focus of cryptoanalysts soon. Anyone still using RSA with key lengths of 2048 bits or less needs to take action, urgently.

In this article, I create a key pair with the use of elliptic curves (elliptic curve digital signature algorithm, ECDSA). To find out more about elliptic curve cryptography, please take a look at the English-language paper by the Germany Federal Office for Information Security (BSI) [1]. The type of key pair, whether ECDSA or ED25519, and therefore the choice of curve, has practically no influence on security. Both methods are considered to be very secure; ED25519 delivers slightly better performance under certain conditions.

To begin, generate a key pair for test purposes (Figure 1), upload the public key to the target system, and log in with the new key pair:

```
$ ssh-keygen -t ecdsa -b 384 ➘
             -f ~/.ssh/ecdsa_2024-03
$ ssh-copy-id ➘
  -i ~/.ssh/ecdsa_2024-03.pub thomas@pihole
$ ssh -i ~/.ssh/ecdsa_2024-03 thomas@pihole
```

Having a private key and password to match boosts the level of security. After logging in, you need to edit the `/etc/ssh/sshd_config` configuration file for the SSH daemon on the target system. You will also want to disable SSH login for root with the `PermitRootLogin no` option and disable the password authentication option with the `PasswordAuthentication no` option. SSH keeps active connections open when the server restarts, which means you can run

```
systemctl restart sshd
```

to restart the SSH daemon without locking yourself out. At this point, open another SSH connection as a test to make sure the public key authentication-based login works. This single step,

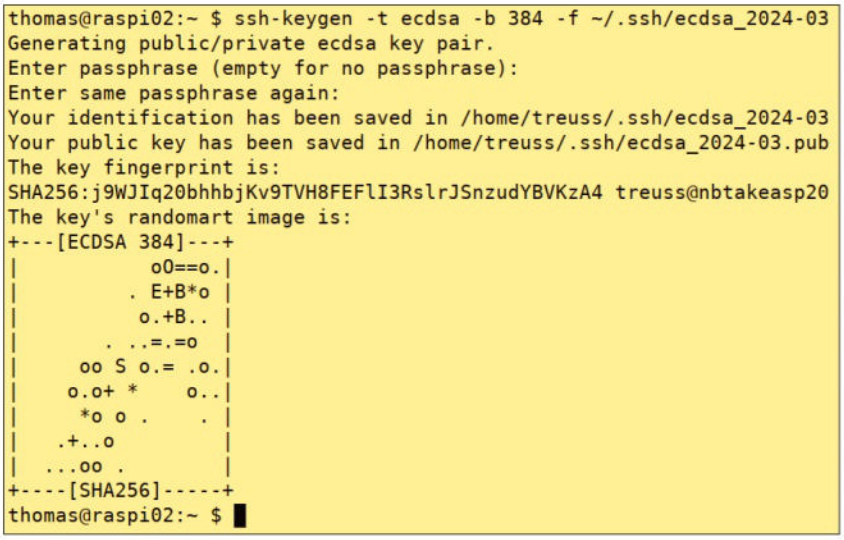
```
$ ssh ➘
  -o PreferredAuthentications=password ➘
  -o PubkeyAuthentication=no root@pihole
```

checks whether root login and password-based login still work.

The Second Factor

Public key authentication is an important step. However, a compromised key pair means that unauthorized persons might be able to gain access to the system. Anyone in possession of the private key can try to unlock it; there is no such thing as a separate, second factor. Many of you will be familiar with the time-based one-time password method (TOTP) from Google Authenticator. The sender and recipient initially agree on a shared secret key; after a defined period of time, typically 30 seconds, a cryptographic hash, the one-time password (OTP), is

Lead Image © shutter999, 123RF.com



```
thomas@raspi02:~ $ ssh-keygen -t ecdsa -b 384 -f ~/.ssh/ecdsa_2024-03
Generating public/private ecdsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/treuss/.ssh/ecdsa_2024-03
Your public key has been saved in /home/treuss/.ssh/ecdsa_2024-03.pub
The key fingerprint is:
SHA256:j9WJIq20bhhbjKv9TVH8FEFlI3RslrJSnzudYBVKzA4 treuss@nbtakeasp20
The key's randomart image is:
+---[ECDSA 384]---+
|           oO==o.|
|         . E+B*o |
|          o.+B.. |
|       . ..=.=o  |
|     oo S o.= .o.|
|    o.o+ *    o..|
|     *o o .    . |
|   .+..o         |
|  ...oo .        |
+----[SHA256]-----+
thomas@raspi02:~ $ █
```

Figure 1: Creating an SSH key pair for test purposes.

computed on the basis of the secret key and the absolute time.

On Debian and its derivatives, you only need one additional package:

```
sudo apt install ➘
  libpam-google-authenticator
```

You then need to tell the system to request the second factor after a successful authentication by the public key method. Here is where the idea of the pluggable authentication module (PAM) comes into play. As the name suggests, PAM can control authentication to certain services with extensible rulesets. If you get something wrong, you might not be able to log in on the system, so it makes sense to make a backup copy of all files in advance.

After integrating the Authenticator library, enable it in PAM by opening the `/etc/pam.d/sshd` file in your choice of editor and adding the line:

```
auth required pam_google_authenticator.so
```

Also make sure the challenge-response procedure is enabled in the `/etc/ssh/sshd_config` file. The `ChallengeResponseAuthentication yes` line is mandatory for TOTP. Finally, restart the SSH daemon by typing:

```
sudo systemctl restart sshd.service
```

You now need to set up Google Authenticator; simply launch `google-authenticator` in a terminal and answer the questions as follows:

```
$ google-authenticator
Make tokens "time-base": yes
Update the .google_authenticator file: yes
Disallow multiple uses: yes
Increase the original generation ➘
  time limit: no
Enable rate-limiting: yes
```

After doing so, you will see a QR code that you can scan with Google Authenticator, as well as some emergency codes.

In principle, the QR code should also work with other TOTP programs, such as KeePassXC. In this case, you need to copy the secret key instead of the QR code. TOTP can be set up in the context menu of the password entry in question. You need to enter the secret key in the matching text box.

The remaining options are just as easy to set up. Use the RFC 6238 option to set the algorithm to SHA-1, the period to 30 seconds, and the code length to six characters. Then, simply enter the TOTP code currently displayed to complete the setup for a genuine second factor on top of your private key.

Highest Hurdles

Hardware tokens seem to be disproportionately widespread among Linux users – at least that was the result of a (not entirely representative) survey at this year's Chemnitz Linux Days. Anyone in possession of a FIDO2 stick [2] can use it for SSH authentication – provided SSH, at least version 8.2p1 or preferably version 8.3, is available on the target system. If you want to use discoverable credentials (formerly known as resident keys), you need SSH 8.3 or later. Up to and including SSH 8.2p1, users are restricted to non-discoverable credentials (previously referred to as non-resident keys).

The difference between discoverable and non-discoverable credentials essentially lies in the storage locations and the associated mechanisms. Discoverable credentials are stored directly on the authenticator itself; that is, they reside there. They can include security tokens such YubiKeys [3], but also Apple Secure Enclaves on the iPhone, hardware security modules (HSMs) on Android devices, or Trusted Platform Modules on a laptop.

The term "discoverable credential" is so called, because the client can determine a list of possible keys in the authenticator that matches the respective relying party ID (rpID), which could be an email address, phone number, or username. Autocomplete only works with discoverable credentials.

Figure 2 shows the login procedure by discoverable credentials. The relying party (RP; e.g., a website) sends a nonspecific authentication request to the client, such as a browser. The client queries the authenticator, a YubiKey in this case, which determines all discoverable credentials for the appropriate RP. In the web browser, you then select the desired discoverable credentials, which in turn are used to sign the request of the requesting party.

This example demonstrates one of the benefits of discoverable credentials. Now that the authenticator includes the relevant user handles (email addresses, usernames, phone numbers, etc.), you no longer need

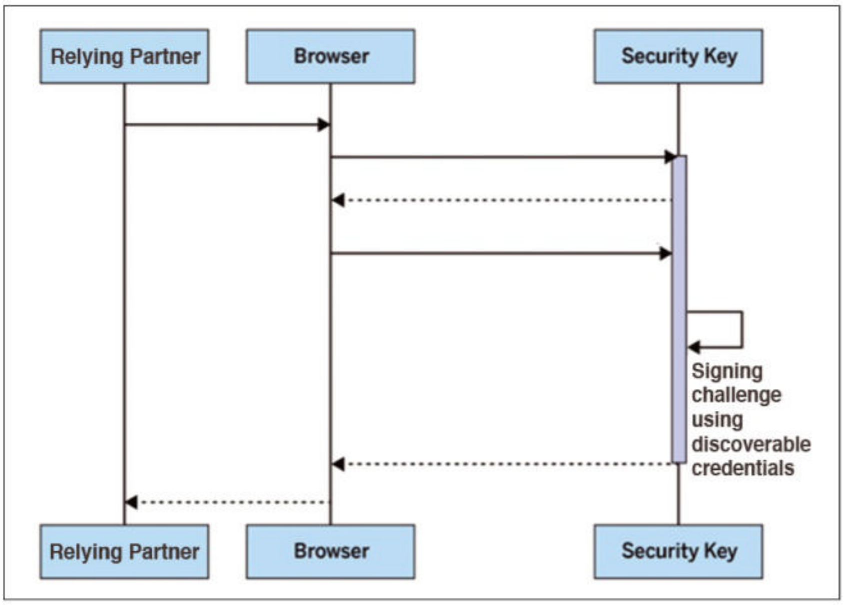


Figure 2: The schematic workflow of an authentication procedure by discoverable credentials. This method means you don't have to memorize your credentials.

to remember them. The handles can be pre-filled. Authentication can also be tied to a specific device (device-specific authentication). However, the process has its downsides because, even in this age of virtually unlimited storage space, many security keys can only store a very limited number of credentials. As a rule, hardware tokens offer between eight and 100 storage slots for resident keys. If you reach the limit, the content in some of the memory slots must be cleared to create space for new credentials.

Another risk is losing the hardware token. If this happens, all the stored keys are lost, which might mean that you can no longer log in to the services you need. Moreover, you run the theoretical risk that an attacker could manage to extract the stored credentials from a captured security key. Although modern authenticator devices are hardened to prevent this outcome (e.g., mechanisms such as PINs or biometric data to prevent extraction), there is always a residual risk.

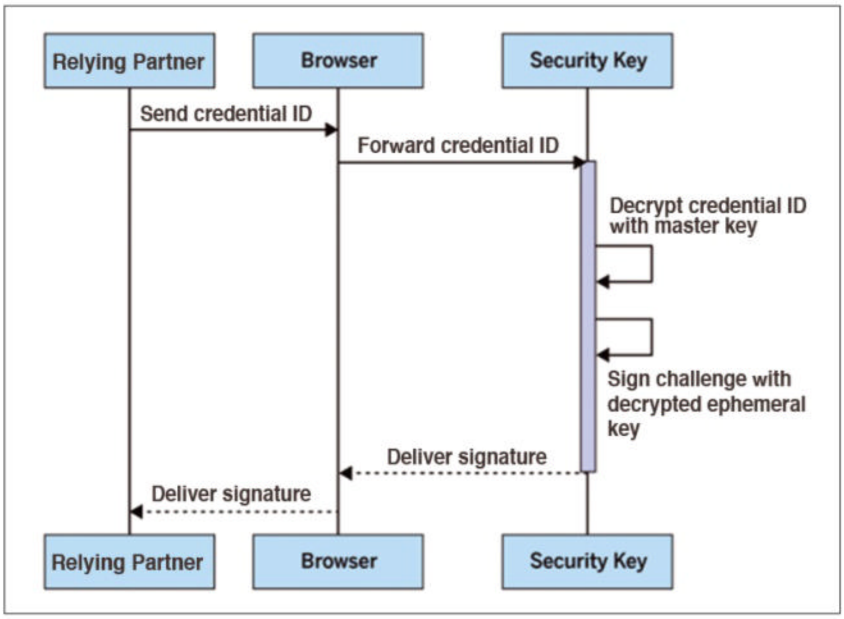


Figure 3: The schematic workflow of an authentication procedure by non-discoverable credentials, which the authenticator itself cannot identify.

Non-Discoverable Credentials

Non-discoverable credentials are a slightly different beast. In contrast to the resident keys stored on the device, these credentials are not stored on the hardware token. Instead, the authenticator generates a separate key pair for each requester based on the protected, internal master key. Its public key is sent to the RP's requesting server, along with a credential ID. The RP then maps this public key with the user account. For each subsequent authentication, the authenticator uses the credential ID to derive the private key from the master key. Because this key is stored only temporarily in the authenticator's protected memory, it is called an ephemeral key. Importantly, the authenticator itself cannot identify the non-discoverable credentials (Figure 3). It cannot search for the key from the rpID and, without the credential ID, would not even be able determine whether a matching key exists at all.

Obviously, in the case of non-discoverable credentials, the RP first needs to obtain the email address, username, or telephone number. After doing so, it can send the associated credential ID back to the browser. As soon as the security key receives this credential ID from the browser, it can derive an ephemeral key by way of the master key and use it for signing.

In contrast to the discoverable credentials, you find no restrictions. The keys are not stored on the authenticator, which means you can basically have any number of keys. If you use non-resident keys, you are also not tied to specific devices. With a suitable authenticator, authentication works across any number of devices and platforms.

However, this greater flexibility also comes with a downside: You need to remember which user handle (email, username, phone number) you used for which website. Additionally, websites need to store the links reliably between user accounts and public keys. A lack of diligence can all too easily lead to serious security problems.

The question is: What does this means in relation to SSH authentication? Third parties cannot use non-discoverable credentials if they do not have the associated credential ID file – even if they know the PIN. This procedure is ideal for environments in which confidentiality must be guaranteed even if a YubiKey is lost.

In contrast, discoverable credentials are impressive in terms of their flexibility. The only way to log in from an arbitrary workstation is to touch the YubiKey and enter the FIDO2 PIN. If the PIN is known, this procedure is ideal when you require particularly simple processes.

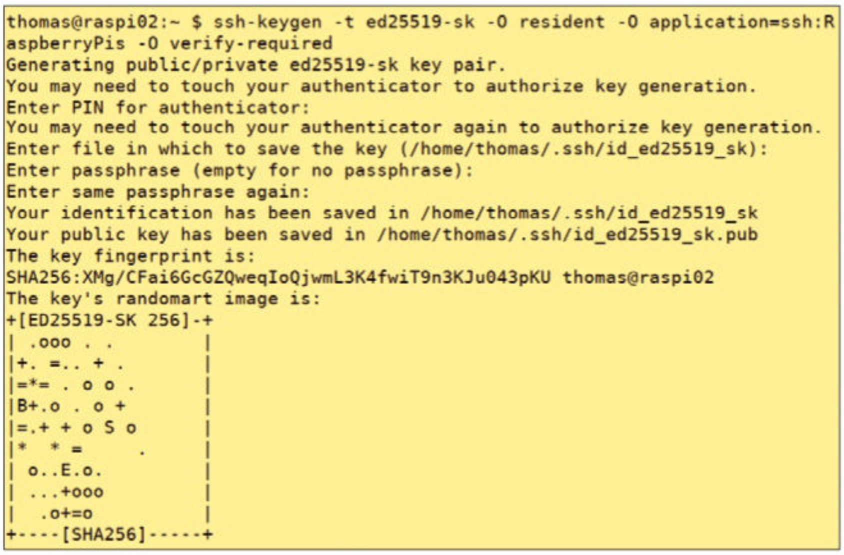
OpenSSH and FIDO2

To set up OpenSSH with FIDO2 authentication, you need the right OpenSSH version, as mentioned earlier. A PIN for FIDO2 must also be set on the YubiKey. Once these requirements are met, you need to add a line for the `PubkeyAuthOptions verify-required` option to the `/etc/ssh/sshd_config` file on all the remote systems and then restart SSH. You can restrict this to specify credentials by adding `verify-required` as a suffix to the matching entries in `~/.ssh/authorized_keys`.

To use discoverable credentials with a YubiKey, first insert the token; then, load a key pair on the basis of the ECDSA curve (first command) or ED25519 curve (second command):

```
$ ssh-keygen ↩
  -t ecdsa-sk ↩
  -O resident ↩
  -O application=ssh:<description> ↩
  -O verify-required
$ ssh-keygen ↩
  -t ed25519-sk ↩
  -O resident ↩
  -O application=ssh:<description> ↩
  -O verify-required
```

You need firmware version 5.2.3 or later on the YubiKey.



```
thomas@raspi02:~ $ ssh-keygen -t ed25519-sk -O resident -O application=ssh:R
aspberryPis -O verify-required
Generating public/private ed25519-sk key pair.
You may need to touch your authenticator to authorize key generation.
Enter PIN for authenticator:
You may need to touch your authenticator again to authorize key generation.
Enter file in which to save the key (/home/thomas/.ssh/id_ed25519_sk):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/thomas/.ssh/id_ed25519_sk
Your public key has been saved in /home/thomas/.ssh/id_ed25519_sk.pub
The key fingerprint is:
SHA256:XMg/CFai6GcGZQweqIoQjwmL3K4fwiT9n3KJu043pKU thomas@raspi02
The key's randomart image is:
+[ED25519-SK 256]-+
| .ooo . .        |
|+. =.. + .       |
|=*= . o o .      |
|B+.o . o +       |
|=.+ + o S o      |
|*  * =      .    |
| o..E.o.         |
| ...+ooo         |
|  .o+=o          |
+----[SHA256]-----+
```

Figure 4: **Creating an ED25519 key pair.**

The `<description>` is simply text that describes where the key is used (e.g., a server name) to help users identify the correct discoverable credential if several are stored on the YubiKey. Although a description is optional, I highly recommend entering a meaningful text.

Although the conventional key pair I created with `ssh-keygen` in Figure 1 still exists, the tool in Figure 4 generates a variant optimized for security keys. You need to both enter a PIN and touch the authenticator.

After entering `ssh-copy-id` to transfer the public key to the target system, the next step is to check that the login works. You have to prove that you are physically present (i.e., touch the YubiKey). The SSH client then prompts for the PIN. If everything works, SSH confirms that you are present and lets you access the system.

Conclusions

If you plan to open access to SSH (possibly even to users on the Internet), you need to harden your authentication process. Both Google Authenticator (i.e., TOTP) and hardware-supported authentication by secure key offer huge security benefits with a manageable configuration overhead. ■

Info

[1] Elliptic curve cryptography: [https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TR03111/BSI-TR-03111_V-2-1_pdf.pdf?__blob=publicationFile&v=1]

[2] FIDO Alliance documentation: [https://fidoalliance.org/specifications/]

[3] YubiKey: [https://developers.yubico.com/SSH/Securing_SSH_with_FIDO2.html]

Automated health checks

Vital Signs

The open source Dradis framework helps you create plans for carrying out team pentests and facilitates the task of standardizing reports from different tools to create summary output. By Matthias Wübbeling

In most cases, the basis for collaboration is good communication and documenting processes, events, and results. Today, you have access to countless tools and frameworks for this process, often specialized for particular kinds of work. London-based Security Roots is the developer of the open source Dradis [1] software for IT security teams. The framework creates standardized reports specifically for security checks, helps teams prepare for penetration testing of IT infrastructures, and organizes the implementation and evaluation.

Security experts often use an expansive kit of tools, each with its specific focus, when carrying out penetration tests. Although some of these tools support standardized output formats for the results, the penetration tester is then ultimately forced to compile and organize things on their own to create a comprehensive report for all the tests. Because no uniform standards exist for organizing or creating reports from the individual results, the developers at Dradis stepped in with a web application that acts as a central interface for the penetration testing process.

The free community version allows several employees to work on one project per instance. You can use various plugins to provide data from common penetration testing tools within the scope of the project, including add-ons for Metasploit, Nessus, Nikto, and Nmap.

First Steps in the Container

Of the various installation options for viewing Dradis in action, I'll first take a quick look at the Docker image:

```
docker run -it --rm ↩
          -p 3000:3000 dradis/dradis-ce
```

Of course, you can also download sources from the Git repository [2] and install the software on your local system. If you have access to Heroku or DigitalOcean, you can install Dradis there directly from the Git documentation at the push of a button.

If you now type *http://localhost: 3000* in your web browser's address bar, you are first prompted to define a password. In the Community Edition, this is the team password, which all team members then use to authenticate for access to the project. If you want to have a look around, populate the setup with sample data as your second step by selecting the *No, I'm a new user* button.

As soon as the sample data import completes, you are redirected to the login page, where you can select an individual username and log in with the team password you selected previously. In my tests, I always had a session timeout error on the first attempt. Simply try again straightaway and be patient, because it will take a moment to call the dashboard for the first time. When done, you will see a summary of current issues, the progress of the project, and the individual activities for the sample project.

Kanban Board

If you click the *Getting started with Dradis Checklist* link in the middle of the project step, you are taken to a simple Kanban board in the *Methodologies* item in the sidebar, where you model the systematic process of your penetration testing as a methodology, which is then processed in the form of individual cards on the board.

If you need further lists for your own workflow (e.g., a *Backlog* or a *Blocked/Need input* list) for work that cannot be continued at the moment, you can easily create them.

To call up edit mode, where you can assign a task to a member of your team, select a card and press *Edit*. Although this step can help with planning, with no overview of the tasks assigned to you, you will not find it very useful when implementing the tests. Overall, the methodology boards offer the basic functions, but you don't expect to find the same

Lead Image © kritiya, 123RF.com

feature scope as in Trello. Nevertheless, it is worthwhile taking the time to work through the cards one after the other to dig down into Dradis.

Editing Findings

For the next step in this overview, select *All issues* from the left sidebar, which takes you to a table view of the issues in the system (i.e., the findings from your penetration test enriched with recommendations for action for the responsible administrators). To change the default set of fields in the table, you can click on the small arrow next to *Columns* icon and select, say, *Description* as an additional column. Significantly more information for the individual entries is displayed. You can use tags to rate the criticality of your findings in Dradis. Some of these are preconfigured, but you can also create your own tags.

Click on any entry to view more detailed information on a finding, such as the one for Apache server version 1.3. The documentation contains detailed information and suggestions for solving the problems. In the *Evidence* tab, you will see evidence to support the findings – typically log data from the penetration testing tool with a reference to the network nodes on which the vulnerability exists. For the sample data populated for new users, the web server on 10.0.155.160 is affected.

The tabs for *CVSS* and *DREAD* allow further specification of ratings for the severity of vulnerability in line with these specifications. Now go back to the first tab and scroll all the way down. In the footer you will see that the report was not typed by an analyst, instead the Qualys upload plugin is listed as the author – that is, the report was imported directly from the tool's output.

Creating Findings

In the next step, try out the upload plugins for creating issues. The various Dradis extensions help you import results from external tools into your penetration test. If you do not have a suitable result, just download the sample result from Burp Suite [3]: You just need to save the page as an HTML file; then, click on *Upload* in the menu at top right to open the Upload Manager (Figure 1).

Now select the *Dradis::Plugins::Burp::Html* plugin in step 1; leave the issue's draft status by selecting *Draft* in step 2, and then select the previously downloaded file in step 3.

After the upload, you can monitor the import progress in the output console. When done, select *All issues* again, and you will see the sample data for *grandjuice.store*. As an analyst, you would now process the data and finalize for your own report. Set the status to *Ready for Review* when saving, which means that the report can be published after a quality check by another member of staff.

Generating a Report

Once you and your colleagues have entered all the findings, it's time to generate the final report. You can do this with the Export Manager, which you can access from the Export link in the menu at the top. The default is an HTML export that is based on one of the two ready-made Dradis templates.

Of course, with a little HTML knowledge, you can create your own templates and make them available in the Dradis `./templates/reports/html_export` folder. When I clicked on *Export* for the sample run, my instance complained about not having sufficient access privileges to write the report. This warning seems to indicate a bug in the Docker image. Use `docker ps` to discover your container's name and then solve the problem with the command:

```
docker exec -ti -u root <containername> ➘
  chown rails /app/app/views/tmp
```

You will then be able to export the report without an error message. The practical ability to export to Word and Excel file formats is reserved for users of the Pro version. Armed with your own HTML templates, though, you can achieve a similarly professional look when completing your report.

Conclusions

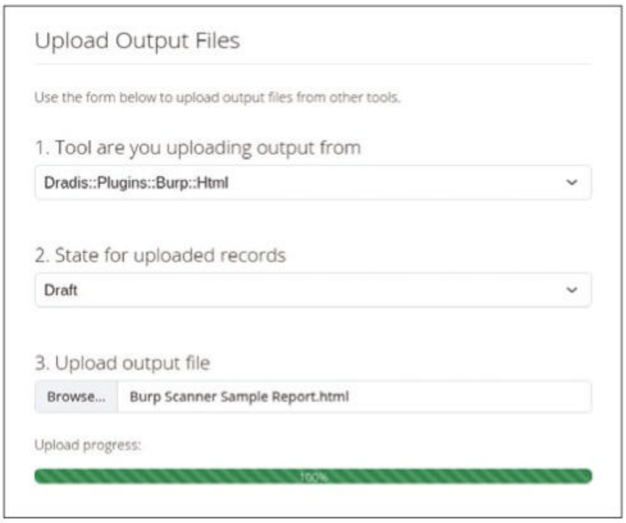
In this article, I provided insights into the basic use of Dradis. Even if the Community Edition is a little limited in terms of functionality in some respects, it is definitely suitable for the team-based preparation, implementation, and reporting of penetration tests. In fact, you will find more use cases for Dradis and, with a little programming overhead, be able to develop additional plugins to import report data from the applications you regularly use, along with templates for exporting your final reports. With the use of different instances, courtesy of Docker and the like, you can implement multiple projects, as well. ■

Info

[1] Dradis: [https://dradis.com]

[2] Dradis repository: [https://github.com/dradis/dradis-ce]

[3] Sample results for Burp Suite: [https://portswigger.net/burp/samplereport/burpscannersamplereport]



Upload Output Files

Use the form below to upload output files from other tools.

1. Tool are you uploading output from

Dradis::Plugins::Burp::Html

2. State for uploaded records

Draft

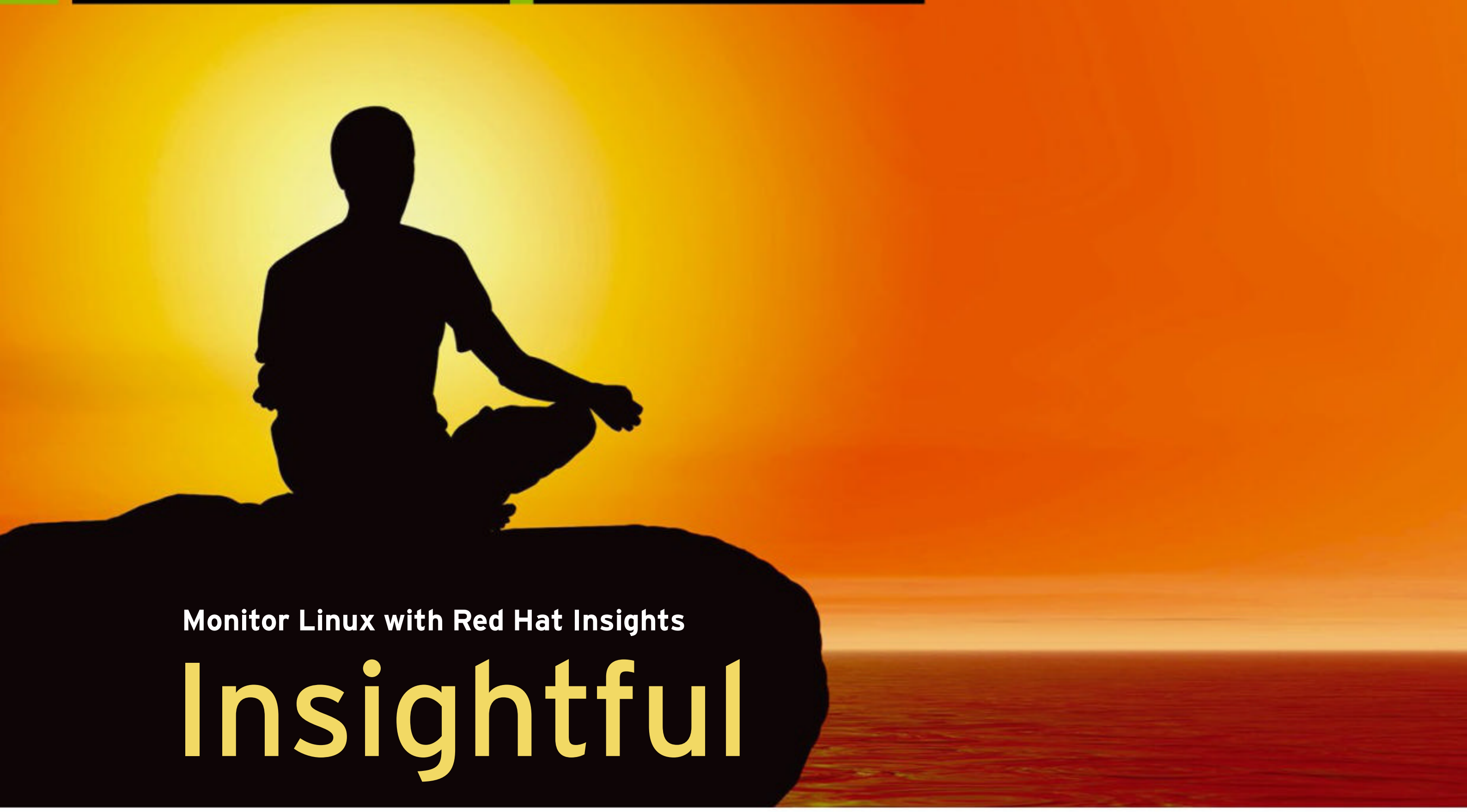
3. Upload output file

Browse... Burp Scanner Sample Report.html

Upload progress:

100%

Figure 1: Downloading a sample result with the Dradis Upload Manager.



Monitor Linux with Red Hat Insights

Insightful

Red Hat's Insights cloud service helps you monitor the security, performance, and availability of Linux systems in hybrid cloud environments; new components now let you create systems with different cloud providers. By Thorsten Scherf

Managing Linux systems centrally in a web interface is old hat rather than Red Hat, who has the Satellite server [1] in its portfolio to perform this function. With this software, you can map the complete life cycle of a system, from initial provisioning through the distribution of software updates and configuration files. Red Hat Insights [2] now offers a cloud service that takes the system monitoring and management approach one step further, especially in hybrid cloud environments.

The service regularly analyzes Red Hat Enterprise Linux (RHEL) systems running the Insights client and proposes actions designed to eliminate potential configuration issues. Insights draws on various databases, and up-to-date common vulnerability and exposure (CVE) and error reports to do this. The databases contain both OpenSCAP-based rulesets for implementing compliance requirements and rulesets that were previously only available in the form of Red Hat Knowledgebase articles [3] (Figure 1).

In other words, to benefit from all the technical expertise, you need to analyze your systems thoroughly on a regular basis. Where appropriate, Insights draws on Ansible playbooks to eliminate automatically the problems it identifies.

Knowledgebase articles are typically authored by Red Hat support engineers in the context of support tickets. The articles usually describe a problem that has been identified on a customer system and put forward concrete solutions for resolving it. Admins who have similar difficulties can easily follow the proposed solution path themselves by referring to the problem description. Let's look at a Red Hat Knowledgebase article on network configuration [4] as a concrete example of how Insights can help you detect potentially incorrect configuration settings.

If a system with the problem described in the Knowledgebase article is registered with Insights, the service would proactively report that the error exists on the system and directly propose potential solutions. Using Ansible Playbooks, Insights can then fix the problem.

Available for Hybrid Cloud Console

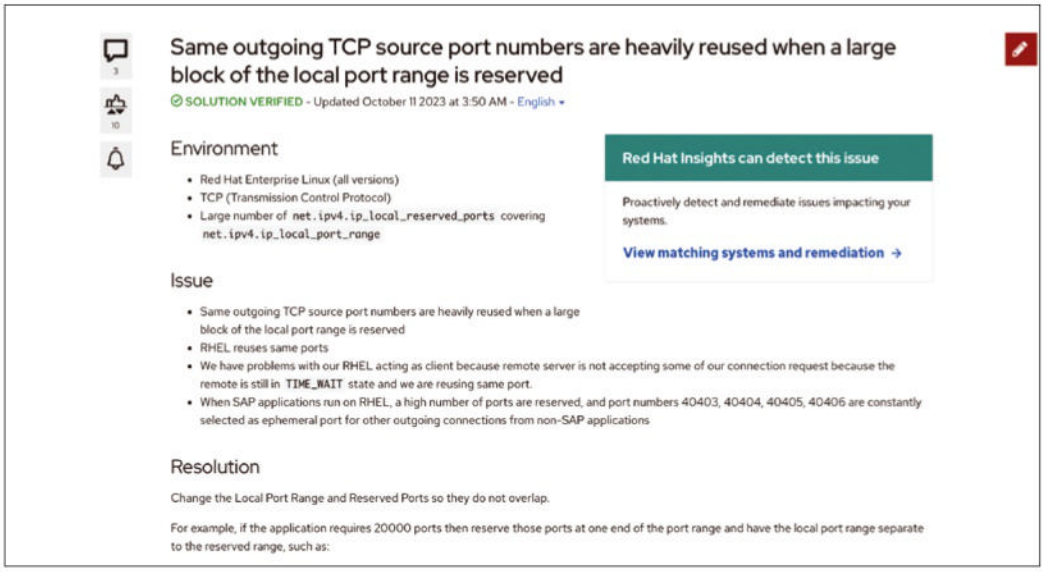
Insights is available as a service for the web-based Red Hat Hybrid Cloud Console [5], for which you need a Red Hat account. You can then use the console to manage your entire infrastructure of RHEL systems, OpenShift clusters, and the Ansible automation platform. All the components in this infrastructure have Insights clients, which means you can use the individual Insights functions on RHEL, the Ansible automation platform, and in OpenShift clusters.

Although you have various ways to register a system with Insights, in this article I just look at RHEL systems and, more specifically, discover how to register a system running the Insights client software with the cloud service. If you have already registered a system with the Red Hat Subscription Manager (RHSM), you just need to install the *insights-client* package and then call up the tool:

```
insights-client --register
```

This command is the method of choice if you have already registered a large number of systems with Red

Lead Image © Rgbspace, Fotolia.com



Same outgoing TCP source port numbers are heavily reused when a large block of the local port range is reserved

✓ SOLUTION VERIFIED - Updated October 11 2023 at 3:50 AM - English ▾

Environment

- Red Hat Enterprise Linux (all versions)
- TCP (Transmission Control Protocol)
- Large number of `net.ipv4.ip_local_reserved_ports` covering `net.ipv4.ip_local_port_range`

Red Hat Insights can detect this issue

Proactively detect and remediate issues impacting your systems.

View matching systems and remediation →

Issue

- Same outgoing TCP source port numbers are heavily reused when a large block of the local port range is reserved
- RHEL reuses same ports
- We have problems with our RHEL acting as client because remote server is not accepting some of our connection request because the remote is still in `TIME_WAIT` state and we are reusing same port.
- When SAP applications run on RHEL, a high number of ports are reserved, and port numbers 40403, 40404, 40405, 40406 are constantly selected as ephemeral port for other outgoing connections from non-SAP applications

Resolution

Change the Local Port Range and Reserved Ports so they do not overlap.

For example, if the application requires 20000 ports then reserve those ports at one end of the port range and have the local port range separate to the reserved range, such as:

Figure 1: **A large number of Knowledgebase articles are available as rulesets in the Insights cloud service.**

Hat with RHMS but have not yet integrated them with Insights.

Activation Key

Alternatively, you can use the remote host configuration (RHC) manager (`rhc` command) to connect new systems to Red Hat Subscription Management and Insights at the same time. As an RHC administrator, you can generate the required activation key in the Hybrid Cloud Console by navigating to *Red Hat Insights* | *RHEL* | *Inventory* | *System Configuration* | *Activation Keys*.

Users of the Red Hat Satellite software will be familiar with the idea of the activation key, which is a simple pre-shared key that lets you register without entering a username and password. You can also bind additional information such as software repositories to the key. This information is then passed on to all systems that you register with the key. To connect a system, run the command:

```
rhc connect ⮐
  -a <Activation Key> ⮐
  -o <Organization ID>
```

At this point, it is also interesting to note that the RHC system consists of a server component as part of the Insights service and a client component on the system you are registering. You can define whether or not the client is allowed to receive and execute changes to Insights configuration files on the console. To do so, navigate to *Insights* | *RHEL* | *Inventory* | *System Configuration* | *Remote Host Configuration (RHCS)*. It makes sense to keep the default settings because Insights will not otherwise be able to resolve potential issues on your systems.

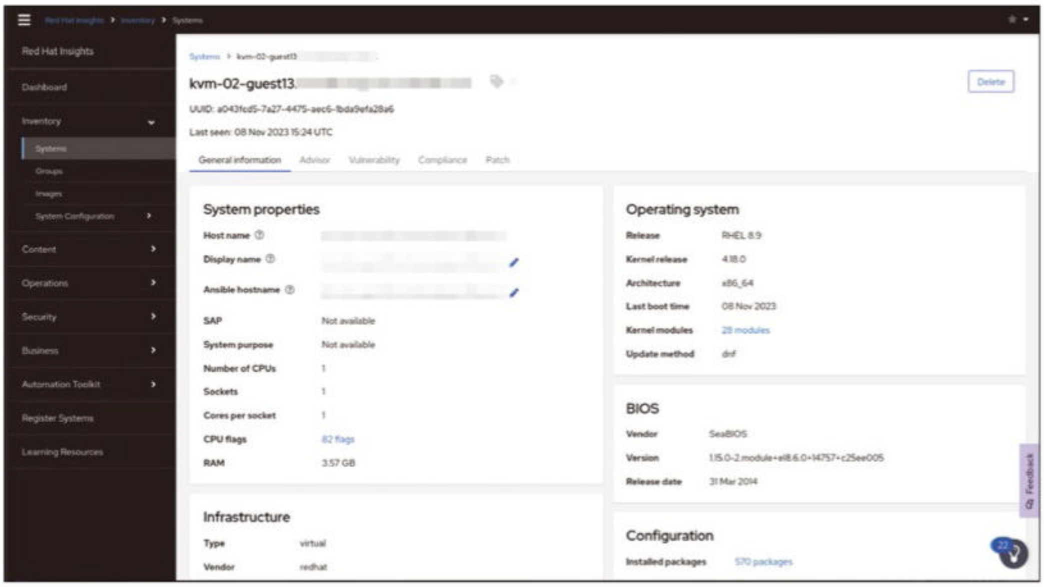
If you have difficulties communicating with the RHC manager, you can use the command

```
systemctl status rhcd
```

to check whether the local client service is running.

Compliance with Insights Advisor

Once you register a system in Insights, it appears in the *Red Hat Insights* | *RHEL* | *Inventory* list on the Hybrid Cloud Console. After selecting the desired system, you will see the *Advisor*, *Vulnerability*, *Compliance*, and *Patch* items, along with some general information about the system (**Figure 2**).



Systems ▸ kvm-02-guest13[redacted]

kvm-02-guest13.[redacted]

UUID: a043fcd5-7a27-4475-aec6-1bda9efa28a6

Last seen: 08 Nov 2023 15:24 UTC

System properties

Host name: [redacted]
Display name: [redacted]
Ansible hostname: [redacted]
SAP: Not available
System purpose: Not available
Number of CPUs: 1
Sockets: 1
Cores per socket: 1
CPU flags: 82 flags
RAM: 3.57 GB

Infrastructure

Type: virtual
Vendor: redhat

Operating system

Release: RHEL 8.9
Kernel release: 4.18.0
Architecture: x86_64
Last boot time: 08 Nov 2023
Kernel modules: 28 modules
Update method: dnf

BIOS

Vendor: SeaBIOS
Version: 1.15.0-2.module+el8.6.0+14757+c25ee005
Release date: 31 Mar 2014

Configuration

Installed packages: 570 packages

Figure 2: **Once a system has been registered, Insights publishes the system analysis results on the web-based Hybrid Cloud Console.**

The Advisor feature delivers information about potential configuration problems on your system. To do this, Insights uses rulesets that are related to the previously mentioned Knowledgebase articles, where you also will find more detailed information about the problem and instructions on how to resolve it yourself.

Alternatively, you can use the *Remediate* button to tell Insights to reference a matching Ansible playbook. To repair the incorrect configuration, the playbook is then automatically executed the next time the system checks in. It is important to note at this point that Insights cannot resolve every single configuration problem with a playbook.

The situation is very similar when you install the Red Hat Security Advisories (RHSAs). In the Vulnerability pane, you can see where you need to install packages on the system to fix the vulnerabilities described in the CVE messages. A list of pending bug fixes (RHBA) or enhancement advisories (RHEA) can be found in the Patch pane. You can use the Hybrid Cloud Console to import all the advisories centrally.

To scan a system for specific compliance requirements, you first need to bind the system to the desired compliance rules. The Insights service gives you an easy option for this purpose. On the Hybrid Cloud Console, navigate to *Security* | *Compliance* | *SCAP Policies* and look for the *Red Hat Insights* | *RHEL* item. If you want to use, say, the Defense Information Systems Agency (DISA) Secure Technical Implementation Guide (STIG) to scan a system, you just add the system in question to this compliance rulebook. You can then configure an automated scan on the system itself at the desired intervals. As an initial test, simply call up the Insights client with the `--compliance` option,

```
insights-client --compliance
```

which immediately runs a scan of the system and sends the results to Insights. When done, you can review the results in the system's Compliance pane.

System Deployment in the Cloud

Over the last few years, the Insights service has added a number of new features. You can now use the image builder tool on the Hybrid Cloud Console to create your own images; you are not restricted to using Red Hat's own software repositories but can also integrate your own repositories, which means you can directly include the software packages you need on your systems when you create a new system image.

To integrate your own software directly into images, Insights lets you set up arbitrary software sources under *Content* | *Repositories*. All you need to do is type in the URL of the repository along with some additional information, such as an optional GPG key to validate the signature of packages originating from this repository.

Insights then enables the deployment of these images in different cloud environments. Connectors are currently available for Amazon Web Services, Microsoft Azure, and the Google Cloud Platform. After provisioning the system, you will automatically have access to all other Insights services.

Creating and Rolling Out Images

Now you can use Insights to create system images, which you can update after deployment in the classic way by `rpm` or `dnf`. Alternatively, immutable images on the basis of OSTree [6] are available and are always used when you want to roll out a completely new version of a system image directly instead of rolling out individual packages. One example is RHEL for Edge systems [7].

Before you create your first image, it makes sense to set up a shortcut between your cloud provider and the Red Hat Hybrid Cloud Console. When creating an image, you can then take the shortcut and use the newly created image directly when provisioning a new system in the target cloud environment.

To create a new shortcut, click on the small wheel at top right on your screen and go to the console *Settings*. Under *Integrations* | *Cloud*, create a new shortcut by clicking the *Add Source* button, select the desired cloud provider, and go through the configuration wizard step by step. Depending on the cloud provider, you will be prompted for various items of information relating to your account. To link to AWS, for example, you will need the Access ID and the Secret Access Key for your AWS account.

After completing these steps, the new shortcut appears in the overview, and you can click it when creating new images to deploy the image within the cloud environment you just configured.

Next, create some images for your cloud environments under *Inventory* | *Images*. As mentioned earlier, you need to decide whether you want to create an RPM/DNF or OSTree-based image. In this article, I use an RPM/DNF image to provision a RHEL system. To launch the wizard, press the *Create Image* button. In addition to the operating system and architecture, here you can specify the desired cloud environment for deploying the image.

In the next step, select the shortcut you set up earlier under *Integrations* so you can use your stored account to access this environment. In the *Register* option, select an activation key with which the new system will be registered in Insights when you launch this image to create a new system.

Once you have decided on a suitable partitioning layout, you can finally select the software packages you want to include in this image. You can access the repositories you bound to the activation key and the custom repositories created previously under *Content* | *Repositories*.

In the final step, you will see a summary of the data you entered. Once you confirm, the image builder starts creating the new image in Insights.

Starting New Systems

As soon as this process is complete, the new image appears in the overview under *Inventory* | *Images*. In the Instance column, you can then select *Launch* to create new systems on the basis of the image you just created. Another wizard launches to guide you through the process step by step. For an AWS image, for example, you can specify the region you want to use for the launch, as well as the types and the number of instances. You can also store an SSH that can be used to log in to the system after a successful launch. Again, you will see an overview of the data you entered when the wizard is done.

The whole process takes a few minutes. Assuming everything went well, Insights shows the *System launched successfully* message, and you can use the `ssh` command listed on the screen to log in to the new system instance with SSH. Because the system was already registered with Insights during the launch process with the activation key mentioned earlier, it is now listed under *Inventory* | *Systems*.

Filtering Out Sensitive Data

Because Insights continuously monitors systems for potential risks and incorrect configurations, the Insights client service regularly sends data to the Insights cloud service – once a day by default. The transferred dataset contains the essential information, which is less data than the typical `sosreport` output of the type typically sent to Red Hat Technical Support in the scope of a support ticket.

The client service encrypts all data transmissions over a TLS-protected communication channel. You can also clean up the dataset to be transmitted in advance to ensure that hostnames and IP addresses are transmitted anonymously. To do so, edit the `/etc/insights-client/insights-client.conf` file and make sure that the following two configuration options are enabled:

```
obfuscate=true
obfuscate_hostname=true
```

If you want to anonymize more data in the dataset, you can do so by creating an `/etc/insights-client/remove.conf` file, which ensures that specific files are excluded from the dataset, the output of certain commands does not become part of the dataset, certain patterns are removed, and specific keywords are anonymized:

```
[remove]
files=/etc/hosts
commands=/bin/dmesg
patterns=password,username
keywords=Secret123,PassW0rd
```

On discovering a pattern in the dataset, Insights removes the entire row in which the pattern appears. The keywords are replaced by a string: `keyword[0]`, `keyword[1]`, and so on.

To make sure data anonymization is working as intended, you can use the `--no-upload` option:

```
insights-client --no-upload
```

The tool then creates a record by using the `remove.conf` file without sending it to the Red Hat Insights Service. You can then unpack the archive locally to check whether the intended changes were made to the files.

Please note that new versions of the Insights Client also support YAML files. Instead of the `remove.conf` file, you can work with two files named `file-redaction.yaml` and `file-content-redaction.yaml`. In contrast to `remove.conf`, these files also support the use of regular expressions.

Conclusions

With Insights, Red Hat provides a cloud service that proactively analyzes RHEL environments to help you identify and resolve configuration and compliance issues quickly. Insights also detects and reports missing patches.

The service now also supports the provisioning of RHEL systems in hybrid cloud environments by helping you create your own images in Insights, which you can then use to deploy the systems. As always, reading the very comprehensive documentation [8] will help you familiarize yourself with the very powerful feature set of the Insights cloud service. ■

Info

[1] Red Hat Satellite server: [https://www.redhat.com/en/technologies/management/satellite]

[2] Insights: [https://www.redhat.com/en/technologies/management/insights]

[3] Red Hat Knowledgebase: [https://access.redhat.com/search/]

[4] Knowledgebase articles: [https://access.redhat.com/solutions/6443541]

[5] Hybrid Cloud Console: [https://access.redhat.com/products/red-hat-hybrid-cloud-console]

[6] OSTree: [https://ostreedev.github.io/ostree/introduction/]

[7] Edge computing: [https://www.redhat.com/en/technologies/linux-platforms/enterprise-linux/edge-computing]

[8] Insights documentation: [https://docs.redhat.com/en/documentation/red_hat_insights/1-latest]

The Author

Thorsten Scherf is the global Product Lead for Identity Management and Platform Security in Red Hat's Product Experience group. He is a regular speaker at various international conferences and writes a lot about open source software.

Networking strategies for Linux on Azure

Blue Skies

We explore advanced networking strategies tailored for Linux workloads on Azure. By Marcin Gastol

An analysis of the critical networking components to optimize the performance and security of Linux workloads on Azure includes a look at virtual networks (VNets), network security groups (NSGs), and custom routing. In this article, I offer practical insights and best practices for IT professionals while delving into performance tuning through Accelerated Networking, traffic management with Azure Load Balancer and Application Gateway, and secure hybrid connectivity with VPN Gateway and ExpressRoute.

Overview

Networking in Azure is a critical aspect of managing Linux-based environments, given the growing prevalence of Linux workloads in enterprise cloud deployments. Azure, Microsoft's cloud platform, provides an array of networking tools that are designed to support the unique needs of Linux applications. These tools allow for the creation of high-performance, secure, and scalable networks, which is important for managing complex, distributed Linux-based workloads.

At the heart of networking in Azure is VNet, which serves as a logically isolated network where Linux virtual machines (VMs) and other services operate. VNets enable IT professionals to establish secure, private networks that can be customized to fit a variety of network topologies. This adaptability is important in Linux-based environments, where the network architecture needs to be both strong and flexible to accommodate the demands of diverse applications.

Azure's networking capabilities are particularly well suited for Linux systems, offering the necessary tools to optimize network performance, security, and reliability. The platform supports advanced configurations such as multiple network interfaces on a single VM, custom routing, and the use of both public and private IP addresses. Additionally, Azure's support for IPv6 and its API offerings allow for deep customization and automation, which are important for maintaining and scaling large Linux deployments efficiently.

Networking Challenges and Opportunities

Deploying the Linux operating system (OS) in Azure presents unique networking challenges, but these challenges also bring opportunities to influence Azure's powerful features to build efficient and secure network infrastructures.

One of the challenges is the complexity of network design and configuration. Creating a network in Azure for Linux environments can be intricate, particularly when integrating on-premises networks with Azure's VNets. The process requires careful management of IP address schemes, routing tables, and subnet configurations to avoid conflicts and ensure efficient traffic flow. Additionally, ensuring the security of these networks against unauthorized access adds another layer of complexity.

Security management is another concern when deploying Linux VMs in Azure. Protecting these VMs requires a security strategy that addresses both external threats and internal vulnerabilities. Misconfigured security rules, particularly in NSGs or Azure Firewall, can lead to the exposure of sensitive data or unauthorized access, compromising the entire network.

Performance optimization is another challenge in managing Linux-based applications in Azure. Ensuring that these applications perform optimally, particularly those that are latency-sensitive or require high throughput, can be difficult. Inadequate network configurations can lead to bottlenecks, negatively affecting application performance and user experience.

Integrating on-premises Linux environments with Azure's cloud infrastructure poses additional challenges, particularly in maintaining consistent security, performance, and management practices across a hybrid setup. Establishing secure connections between on-premises networks and Azure, especially through VPN gateways or ExpressRoute, requires deep knowledge of network engineering and Azure's specific tools.

VNet Architecture

In Azure, VNet is the fundamental building block for networking, providing the framework within which all cloud-based resources communicate.

Lead Image ©pakhnyushchyy, 123RF.com

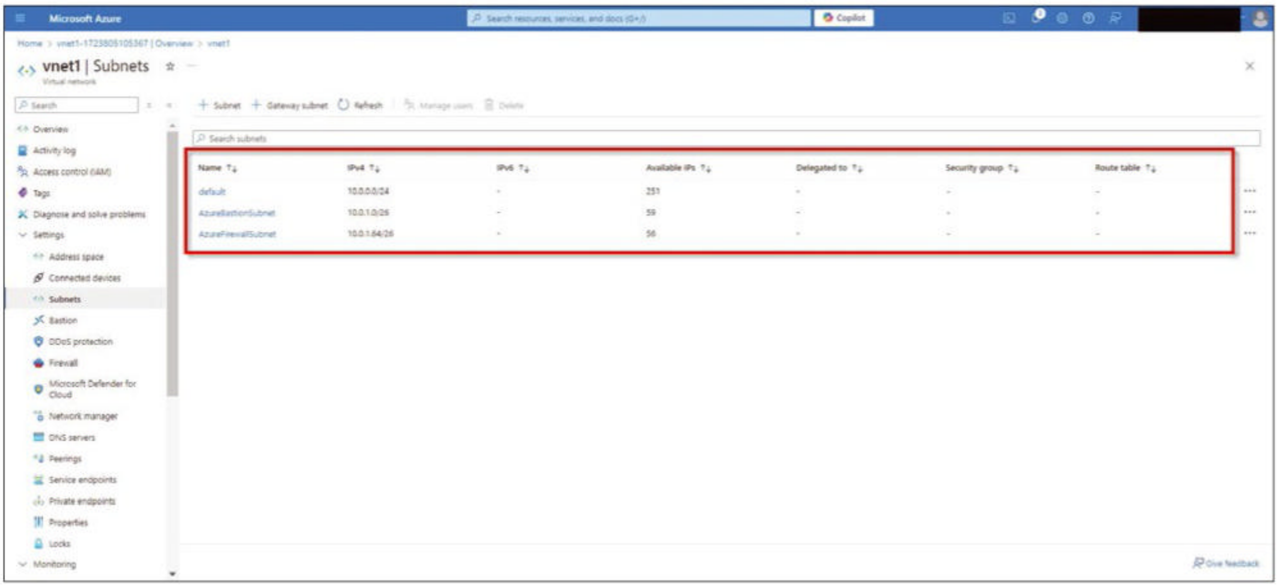
For Linux workloads, a well-architected VNet is important to ensuring secure, efficient, and scalable operations. Advanced VNet design patterns enable IT professionals to create a network architecture that meets the specific needs of Linux VMs, ensuring that these workloads perform optimally and securely within the Azure environment (Figure 1).

Advanced VNet Design Patterns

When designing VNets for Linux workloads, it's important to consider advanced patterns that address the specific requirements of these environments. A typical VNet can be structured to include multiple subnets, each tailored to the different roles that Linux VMs play in your architecture. For instance, separating web servers, application servers, and database servers into distinct subnets not only organizes your network but also enhances security and performance by isolating different layers of your application stack.

One advanced design pattern is the use of hub-and-spoke topologies, wherein a central hub VNet controls communication between multiple spoke VNets. This design is particularly useful in large-scale deployments where you need to manage and secure traffic flow between different Linux environments, such as development, testing, and production. The hub VNet typically contains shared resources, such as firewalls (Figure 2) or VPN gateways, and manages traffic to and from each spoke VNet that hosts the Linux workloads. This pattern simplifies the management of security policies and reduces the complexity of network management.

Another critical design pattern is the use of VNet peering, which allows two VNets to connect seamlessly, enabling direct communication between Linux VMs in different VNets without the need for a VPN or additional gateways. This technique is especially



Name ↑↓	IPv4 ↑↓	IPv6 ↑↓	Available IPs ↑↓	Delegated to ↑↓	Security group ↑↓	Route table ↑↓
default	10.0.0.0/24	-	251	-	-	-
AzureBastionSubnet	10.0.1.0/26	-	59	-	-	-
AzureFirewallSubnet	10.0.1.64/26	-	56	-	-	-

Figure 1: **Subnet configuration within an Azure VNet (vnet1). Three subnets are highlighted: *default*, *AzureBastionSubnet*, and *AzureFirewallSubnet*, configured with specific IPv4 ranges and available IP addresses.**

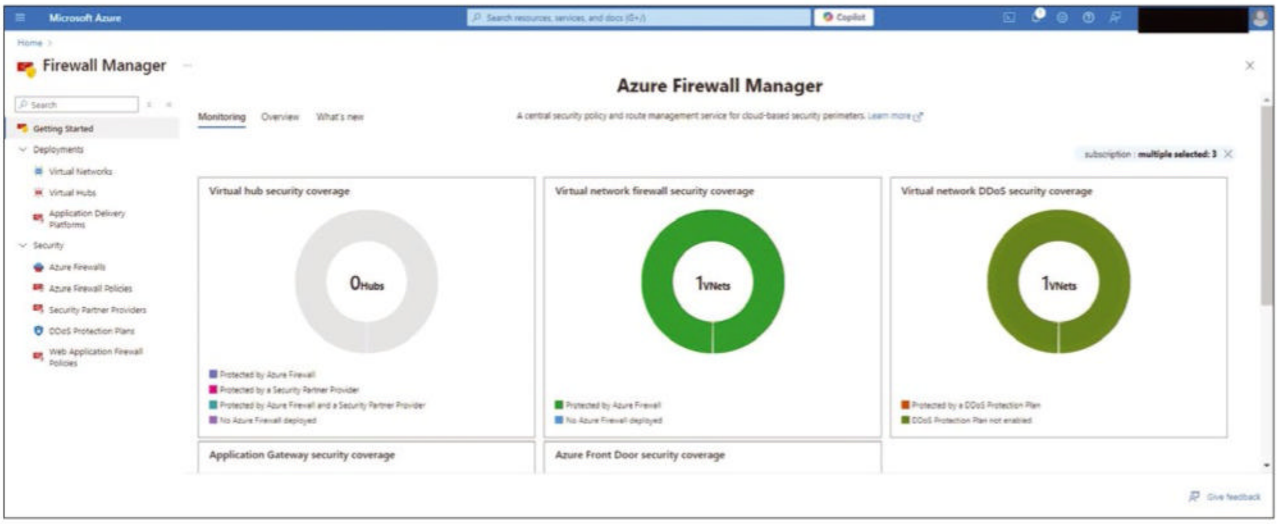

Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
[redacted]
Home >
Firewall Manager
Search
Getting Started
Deployments
Virtual Networks
Virtual Hubs
Application Delivery Platforms
Security
Azure Firewalls
Azure Firewall Policies
Security Partner Providers
DDoS Protection Plans
Web Application Firewall Policies
Azure Firewall Manager
Monitoring
Overview
What's new
A central security policy and route management service for cloud-based security perimeters. Learn more
subscription : multiple selected: 3
Virtual hub security coverage
0 Hubs
Protected by Azure Firewall
Protected by a Security Partner Provider
Protected by Azure Firewall and a Security Partner Provider
No Azure Firewall deployed
Virtual network firewall security coverage
1 VNets
Protected by Azure Firewall
No Azure Firewall deployed
Virtual network DDoS security coverage
1 VNets
Protected by a DDoS Protection Plan
DDoS Protection Plan not enabled
Application Gateway security coverage
Azure Front Door security coverage
Give feedback

Figure 2: **The Azure Firewall Manager dashboard shows an overview of security coverage across your Azure environment.**

useful for multiregion deployments in which workloads need to communicate across regions with minimal latency. VNet peering is cost-effective and provides high-bandwidth, low-latency connections, making it ideal for Linux workloads that require fast, reliable interconnectivity across different geographic locations.

Integration

Customizing VNet configurations to meet the needs of Linux workloads involves more than just setting up basic network parameters. It requires a deep understanding of how Linux systems interact with network resources and how to optimize these interactions for performance and security.

One key aspect of customization is IP address management. For Linux VMs, assigning static IP addresses can be critical, particularly for services that require stable endpoints, such as databases or internal APIs. Azure allows for the reservation of static private IP addresses within a VNet, ensuring that your Linux workloads maintain consistent IP addresses, even through reboots or redeployments.

Routing is another area where customization can importantly affect performance and security. Azure's user-defined route (UDR) tables can be used to define custom routes that override Azure's default system routes. For instance, you might create routes that force all outbound traffic from Linux VMs to pass through a network virtual appliance (NVA) for inspection and logging, thereby enhancing security. Alternatively, you could define routes that optimize traffic flow between VMs across different subnets or VNets, reducing latency and improving performance.

In some cases, you might need to integrate Linux workloads with on-premises networks, for which Azure provides several options. Among these options are VPN Gateway and ExpressRoute, which offer secure and reliable connections between Azure VNets and your on-premises infrastructure. When configuring these connections, it's important to consider factors such as bandwidth, latency, and failover capabilities, ensuring that your Linux workloads can communicate seamlessly with on-premises resources.

Managing NSGs

NSGs play a important role in protecting Linux workloads in Azure by controlling inbound and outbound traffic to VMs according to predefined security rules. Properly configuring NSGs is critical to maintaining the security of your Linux environments.

When managing NSGs for Linux VMs, it's important to implement a least privilege approach, which means only allowing the minimum necessary traffic to and from your Linux VMs. For example, you might create NSG rules that only permit SSH access from specific IP addresses or ranges, thus reducing the attack surface. Additionally, for web servers, you could restrict HTTP/HTTPS traffic to only the necessary ports (e.g., 80 and 443) and block all other traffic.

NSGs can be applied at both the subnet and network interface card (NIC) level. Applying NSGs at the subnet level is generally more efficient and easier to manage because it covers all VMs within the subnet. However, for scenarios in which specific VMs require different security rules, NSGs can also be applied directly to the NICs of individual Linux VMs, providing more granular control.

It's also important to audit and update NSG rules regularly (Figure 3) to ensure they remain aligned with your security policies and the evolving threat landscape. Azure provides tools like Azure Security Center and Azure Monitor that can help you track and manage NSG configurations, ensuring that your Linux workloads are always protected against unauthorized access and network threats.

Optimization

Optimizing network interfaces and IP configurations on Azure is important for ensuring that Linux VMs operate efficiently and securely in a cloud environment. Azure provides a range of advanced networking features that allow IT professionals to fine-tune network performance, manage multiple IP addresses, and use public and private IP addresses effectively to meet the specific needs of Linux workloads.

Advanced NIC Configuration

In Azure, each Linux VM is equipped with at least one NIC that connects it to a VNet. However, for more complex scenarios, such as those requiring advanced routing or network isolation, it is often necessary to configure multiple NICs on a single VM. This setup allows a Linux VM to handle traffic from different subnets, making it possible to segregate different types of traffic for enhanced security and performance.

When dealing with multiple NICs, it's important to configure routing tables properly and ensure that the correct traffic flows through the appropriate NIC. For example, you might dedicate one NIC to handle internal traffic within the VNet, while another NIC manages traffic that flows to external networks or the Internet. This separation of traffic not only enhances security but also allows for more granular control over traffic policies and monitoring.

Additionally, configuring multiple NICs can help in scenarios that require high availability or redundancy. For instance, you can set up multiple NICs connected to different subnets or VNets, allowing the Linux VM to maintain connectivity, even if one network path fails. This approach is particularly useful in mission-critical applications where uptime is important.

Multiple IP Addresses

Linux VMs in Azure can be configured to handle multiple IP addresses on a single NIC. This capability is useful for scenarios in which a VM needs to host multiple services, each requiring its own IP address, or when you need to maintain separate communication channels for different applications or customers.

Configuring multiple IP addresses involves adding secondary IP configurations to a NIC. These addresses can be used by applications running on the Linux VM to listen on different IPs, thus providing a more organized and secure way to manage network traffic. For example, a web server might use one IP address for public-facing traffic and another for internal API calls, thereby isolating different types of network traffic.

To handle multiple IP addresses effectively, it is important to adjust the network configuration on the Linux VM itself, which typically involves modifying the network interface settings to recognize and properly route traffic for each assigned IP address. Tools like `ip` or network configuration files such as `/etc/network/interfaces` (on Debian-based systems) or `ifcfg` files (on Red Hat-based systems) are used to configure the IP addresses and ensure they are recognized by the operating system.

In more advanced scenarios, you might also need to configure `iptables` or `firewalld` rules to manage how traffic is handled by the Linux VM. This step ensures that traffic arriving on specific IP addresses is processed by the correct service, adding an extra layer of security and traffic management.

Implementing and Managing IPs

Azure allows Linux VMs to be assigned both public and private IP addresses, each serving different purposes within a network architecture. Properly managing these IPs is important for maintaining security and ensuring that applications are accessible as intended.

Private IP addresses are used for internal communication within a VNet, making them important for services that need to interact with other VMs, databases, or services, without exposing them to the public Internet. In contrast, public IP addresses are used to expose specific services or applications to the Internet, making them accessible from anywhere.

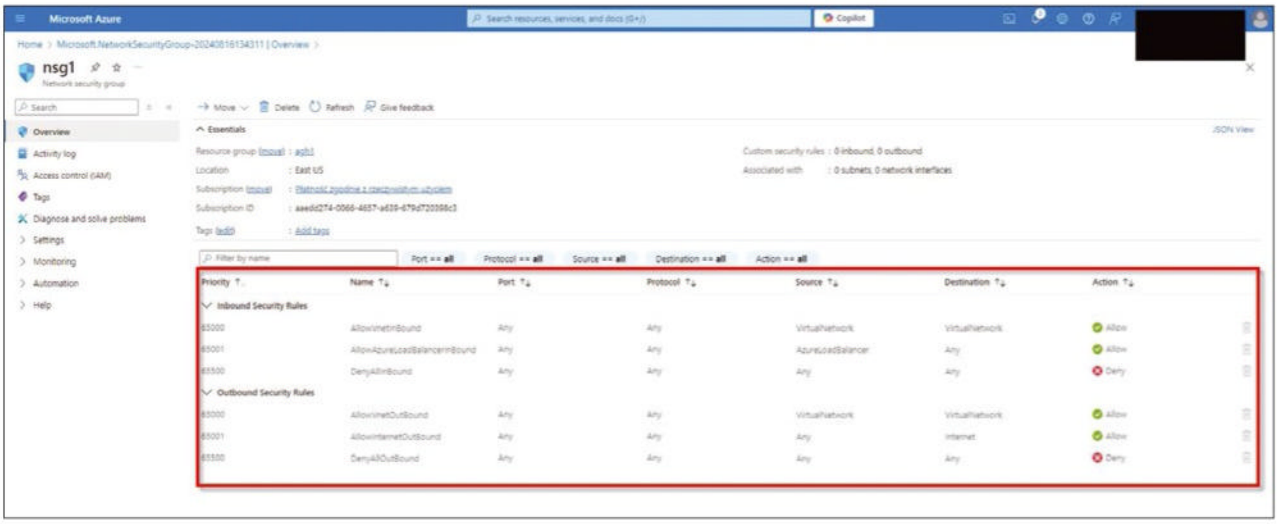
When assigning public IP addresses, it is important to consider security implications. Public IPs should be limited to only those VMs that require external access, such as web servers or VPN gateways. Additionally, the use of NSGs for tight control of inbound and outbound traffic on public IP addresses helps mitigate risks associated with exposure to the Internet.

On the other hand, private IPs should be carefully managed to ensure proper internal communication by making certain IP address ranges do not overlap with on-premises networks (if connected over a VPN or ExpressRoute) and routing is configured correctly to handle traffic between different subnets and VNets. Azure provides features like Private Link and VNet peering to extend the utility of private IPs, enabling secure connections between different Azure services or between Azure and on-premises environments without the need for a public IP.

For scenarios in which high availability and load balancing are required, public and private IPs can be used in conjunction with Azure Load Balancer. For instance, a web application might use a public IP address as the front end for incoming traffic, whereas the back-end VMs communicate by private IPs within a secure VNet. This setup not only enhances security but also improves performance by localizing traffic and reducing the attack surface.

Traffic Management

Effective traffic management is important for maintaining the performance, availability, and security of Linux-based applications hosted on Azure. Azure provides a suite of tools that allows IT professionals to implement strong traffic management strategies, ensuring that applications remain responsive and resilient under varying loads and conditions. Key tools in this suite include the Azure Load Balancer, Azure Application Gateway, and Azure Traffic Manager, each of which plays a distinct role in managing traffic for Linux environments.



[redacted]

Resource group (move) : agh1
Location : East US
Subscription (move) : Płatność zgodnie z rzeczywistym użyciem
Subscription ID : aaedd274-0066-4657-a639-679d720398c3
Tags (edit) : Add tags

Custom security rules : 0 inbound, 0 outbound
Associated with : 0 subnets, 0 network interfaces

Priority ↑	Name ↑↓	Port ↑↓	Protocol ↑↓	Source ↑↓	Destination ↑↓	Action ↑↓
Inbound Security Rules						
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	✓ Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	✓ Allow
65500	DenyAllInBound	Any	Any	Any	Any	✗ Deny
Outbound Security Rules						
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	✓ Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	✓ Allow
65500	DenyAllOutBound	Any	Any	Any	Any	✗ Deny

Figure 3: NSG settings in Azure detail both inbound and outbound security rules.

Configuring for High Availability

The Azure Load Balancer (Figure 4) is a foundational component for achieving high availability in Linux-based environments. It operates at the transport layer (Layer 4) and is designed to distribute incoming network traffic across multiple VMs within a VNet, ensuring that no single VM becomes a bottleneck or point of failure.

When configuring the Azure Load Balancer for Linux workloads, it's important to consider both internal and external load-balancing needs. Internal load balancing is typically used to distribute traffic among VMs within a private VNet, which is common for back-end services such as databases or internal APIs. External load balancing, on the other hand, distributes traffic from the Internet to VMs within a VNet, which is often used for public-facing web servers or applications.

To set up a load balancer, you define a front-end IP configuration, back-end pool, and load-balancing rules. The front-end IP configuration is the entry point for incoming traffic, which the load balancer then distributes to the VMs in the back-end pool according to the defined rules. These rules determine how traffic is distributed, such as by source IP, protocol, or port.

For high availability, it's important to include multiple VMs in the back-end pool, ideally spread across different availability zones or availability sets. This configuration ensures that if one VM or zone experiences an outage, the load balancer can continue directing traffic to the remaining healthy VMs, maintaining the availability of the application.

Additionally, leveraging health probes is important for ensuring that only healthy VMs receive traffic. Health probes periodically check the health status of each VM in the back-end pool. If a VM fails the health check, the load balancer automatically stops directing traffic to that VM until it is back online, thus preventing service interruptions.

Application Gateway

The Azure Application Gateway is a traffic management tool that operates at the application layer (Layer 7), providing more sophisticated routing capabilities tailored to web-based applications. It is particularly useful for Linux-based web services that require advanced routing, Secure Sockets Layer (SSL) termination, and application firewall capabilities.

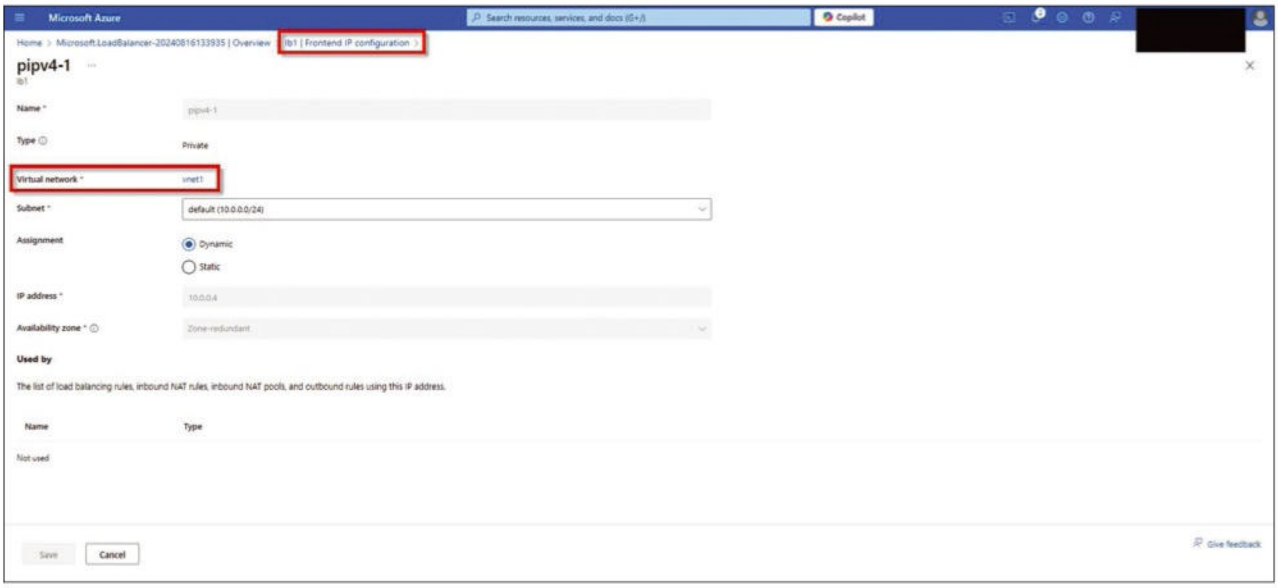
One of the primary features of the Application Gateway is its ability to perform SSL offloading, which reduces the computational load on your Linux VMs by handling SSL termination at the gateway level. This offloading frees up resources on the VMs, allowing them to handle more application-level processing. SSL termination also simplifies certificate management because SSL certificates are only applied at the gateway, rather than on each individual VM.

Another critical feature is URL-based routing, which allows the Application Gateway to direct traffic by URL path. This capability is especially useful for microservices architectures or applications with multiple subdomains. For instance, requests to *api.example.com* can be routed to a specific set of Linux VMs optimized for API processing, whereas requests to *www.example.com* are directed to a different set of VMs that handle web traffic.

The Application Gateway also integrates with the web application firewall (WAF), providing an additional layer of security for Linux-based web applications. The WAF protects against common web vulnerabilities such as SQL injection, cross-site scripting (XSS), and other Open Web Application Security Project (OWASP) top 10 threats. This integration ensures that your Linux web services are not only performant but also secure from common attack vectors.

Traffic Routing

Azure Traffic Manager is a global DNS-based traffic routing service that



[redacted]

Home > Microsoft.LoadBalancer-20240816133935 | Overview > lb1 | Frontend IP configuration >

pipv4-1
lb1

Name * pipv4-1

Type ⓘ Private

Virtual network * vnet1

Subnet * default (10.0.0.0/24)

Assignment ● Dynamic ○ Static

IP address * 10.0.0.4

Availability zone * ⓘ Zone-redundant

Used by

The list of load balancing rules, inbound NAT rules, inbound NAT pools, and outbound rules using this IP address.

Name	Type
Not used	

Figure 4: Configuration of a front-end IP for a load balancer in Azure.

enables you to distribute traffic across multiple Azure regions. This service is particularly beneficial for Linux deployments that need to ensure high availability and optimal performance for users across different geographic locations.

Traffic Manager works by directing incoming DNS requests to the most appropriate endpoint according to the routing method you choose. The available routing methods include Priority, Performance, Geographic, and MultiValue. Each method serves a different purpose, allowing you to tailor traffic routing to your specific needs. For instance, the Performance routing method directs traffic to the closest endpoint with the lowest latency, ensuring that users experience the best possible performance regardless of their location. This method is ideal for global Linux applications where user experience is critical. Alternatively, the Priority routing method can be used to implement a primary backup failover strategy, ensuring that traffic is directed to a secondary region if the primary region becomes unavailable.

Traffic Manager also supports Weighted routing, allowing you to distribute traffic across multiple endpoints on the basis of assigned weights. This feature is useful for gradually rolling out updates or balancing loads across different regions to prevent overloading a single data center.

Integrating Traffic Manager with your Linux deployments involves configuring DNS settings to point to the Traffic Manager profile, which then directs traffic according to the specified routing method. This setup provides a resilient and flexible way to manage global traffic, ensuring that your Linux applications remain available and performant even under challenging conditions.

Accelerated Networking and Performance Tuning

Optimizing network performance is important for Linux workloads running on Azure, especially for applications that demand low latency, high throughput, and consistent reliability. Azure offers several tools and techniques to enhance network performance, with Accelerated Networking being a key feature. Coupled with advanced throughput optimization and latency minimization strategies, IT professionals can ensure their Linux-based applications perform optimally in the cloud.

The Accelerated Networking feature provided by Azure importantly improves the network performance of VMs by reducing latency, lowering jitter, and decreasing CPU utilization on the VM. It does so by offloading network processing from the VM's CPU to the underlying hardware,

specifically to a dedicated NIC that supports single-root I/O virtualization (SR-IOV).

To enable Accelerated Networking on Linux VMs, you first need to ensure that the VM size and the operating system version support this feature. Accelerated Networking is available on most general-purpose and compute-optimized VM sizes in Azure, but it's important to verify compatibility with the specific Linux distribution you are using.

Once compatibility is confirmed, Accelerated Networking can be enabled during the VM creation process or on existing VMs by attaching a supported NIC through the Azure Portal, Azure command-line interface (CLI), or Azure Resource Manager (ARM) templates. For example, from the Azure CLI, you can enable Accelerated Networking on a NIC with the command

```
az network nic update ➘
  --name <nic-name> ➘
  --resource-group <resource-group-name> ➘
  --accelerated-networking true
```

After enabling Accelerated Networking, you should verify that the feature is working as expected by checking the NIC properties in the Azure Portal or by issuing the `ethtool` command on the Linux VM, which should show that SR-IOV is enabled.

Network Throughput Optimization

Network throughput is a critical factor in the performance of many Linux-based applications, especially those that handle large volumes of data or require high-speed data transfer. Azure provides several techniques to optimize network throughput for Linux VMs, ensuring that applications can handle their workloads efficiently.

One key technique is to select VM sizes that are optimized for network performance. Azure offers specialized VM sizes, such as the H-series for high-performance computing or the D-series for general-purpose workloads, that provide enhanced network capabilities. These VMs often come with higher network bandwidth limits and are ideal for applications requiring substantial data movement.

Another throughput optimization technique is to ensure that the underlying storage infrastructure is not a bottleneck. The use of Azure Premium SSDs or Ultra Disks can importantly improve data read/write speeds, which directly affects the network performance of data-intensive applications. Combining these techniques with Accelerated Networking can lead to substantial performance gains.

Configuring the Linux VM's network stack to handle high throughput, which includes tuning TCP settings such as window size, buffer sizes, and congestion-control algorithms, is also important. Tools such as `sysctl` can be used to adjust these parameters, allowing the Linux kernel to manage larger data flows more effectively, for example:

```
sysctl -w net.core.rmem_max=16777216
sysctl -w net.core.wmem_max=16777216
sysctl -w net.ipv4.tcp_window_scaling=1
```

These adjustments help in maximizing the throughput by allowing the Linux VM to process more data packets simultaneously, reducing delays caused by network congestion.

Latency Minimization Strategies

Minimizing latency is important for latency-sensitive Linux applications, such as those used in real-time data processing, financial transactions, or interactive user interfaces. Azure offers several strategies to reduce network latency, ensuring that Linux workloads remain responsive and efficient.

One of the primary strategies is to use Accelerated Networking, which, as mentioned earlier, reduces network latency by offloading processing to the NIC hardware. This method is particularly effective in scenarios where even microseconds of delay can have an important effect on application performance.

Keeping resources close to each other geographically reduces the physical distance data must travel, thereby minimizing latency. Additionally, the use of Azure proximity placement groups can further reduce latency by ensuring that VMs are physically located close to one another within the same data center.

For applications that require global reach, Azure Traffic Manager can be used to direct users to the nearest Azure region by network performance metrics. This approach helps minimize latency for end users by routing their requests to the closest available resource, reducing the time it takes for data to travel across the network.

Finally, optimizing the Linux kernel's networking stack for low latency is important and includes disabling features that introduce unnecessary processing delays, such as TCP slow start and tuning the interrupt coalescing settings on the NIC to reduce the time it takes to process network packets. Tools like `ethtool`,

```
ethtool -C eth0 rx-usecs 0
```

adjust these settings.

Summary

In this article, I explored the critical components of networking for Linux workloads on Azure, from architecting advanced VNet designs and customizing configurations for optimal IP management and routing to securing environments with NSGs and advanced firewall configurations.

The Author

Marcin Gastol is a Senior DevOps Engineer and Microsoft Certified Trainer with extensive experience in Azure technologies. He has taught various IT subjects, hosts a blog that addresses multiple IT areas ([https://marcingastol.com/]), and speaks at various conferences.

Data deduplication on Windows Server 2022

Double Trouble

Data deduplication tools boost efficiency in storage management by improving storage utilization and saving network bandwidth during backup and replication processes. By Thomas Joos

Files are often stored multiple times, unnecessarily hogging storage space on data carriers and in backups; therefore, data deduplication is particularly useful for file servers. Virtual desktop infrastructure (VDI) environments also benefit from this technology, for which data deduplication even offers separate options that I also cover in this article. The technology can use both physical data carriers and virtual disks, which means that deduplication can be used effectively in virtual environments.

Once the feature has been installed, the connected hard drives are checked according to a schedule, and the deduplication rate is displayed in the Server Manager console. In this way, you can keep an eye on the success and utility value of deduplication for individual disks. Because you are not forced to deduplicate every single data carrier on a server, you can flexibly control which data carriers are worth the overhead.

If you discover that deduplication does not deliver meaningful results for individual data carriers, you can remove them from the configuration at any time. Starting with Windows Server 2019, deduplication not only supports NTFS, but also the Protogon resilient filesystem (ReFS), and therefore very large data carriers, as well.

All told, data deduplication on Windows Server 2022 is a powerful way to optimize storage capacities and reduce costs. However, you need to find a balance between the benefits and the potential challenges to ensure smooth and efficient system management. In this article, I shed light on the technical background of deduplication and show you how to set up and manage data deduplication in a graphical interface, with PowerShell, and from the command line on Windows Server 2022.

The Downside

Of course, you also need to consider the potential downsides of data deduplication because it is not useful in all environments. Database servers, Exchange, or Hyper-V hosts will rarely benefit, although VDI environments are an exception. On virtual machines (VMs), in contrast, deduplication can certainly offer benefits, depending on the server role. Virtual file servers benefit from data deduplication just as much as physical file servers. One possible disadvantage is that data deduplication requires compute resources, which can affect performance on servers that are already heavily utilized. Initializing the deduplication process in particular may involve heavy use of CPU and memory.

Moreover, it is important to configure deduplication carefully to make sure important files are not excluded or inadvertently modified. Another aspect to keep in mind is the dependency of data recovery on deduplication. Because deduplicated data is stored once only, recovery can be more complex than with conventional methods, which entails careful planning and regular testing of backup and recovery processes.

When Not To Dedup

After you decide to use data deduplication on Windows Server 2022, you need to remember that this technology is not the answer for some data types and files. In fact, some data types will benefit less or not at all from deduplication. Formats such as JPEG, MP3, MP4, or ZIP, which already use forms of data compression, offer little leverage for reducing redundancy and saving storage space. Active database files, especially files that require frequent write operations, are generally not a good choice for deduplication. Constant changes to them can affect deduplication

Lead Image © iofoto, Fotolia.com

efficiency and, in some cases, performance. Files that are updated in real time, such as system logs, can even be blocked by the deduplication process. Constant write access to these files conflicts with the way deduplication works.

Deduplication is more suited to static data or content that is not frequently modified. Also, where files are individually encrypted, each file is unique, even if the original unencrypted content was identical. This characteristic considerably limits the effectiveness of deduplication, because no significant redundancies can be identified.

Data deduplication is also possible in storage pools and on virtual hard drives. If you have installed role services, a window will appear when you create a new volume. You can use it to enable deduplication for the current volume, assuming deduplication will work on it. It does not matter whether you use data deduplication for data on normal volumes or on virtual disks in storage pools.

SSD, HDD, and NVMe

When implementing data deduplication in environments that use different storage technologies (e.g., hard disk (HDD), solid-state (SSD), and non-volatile memory (i.e., NVMe) drives), you need to consider several aspects. Data deduplication performance can vary greatly depending on your choice of storage technology. HDDs with their slower access times could lead to bottlenecks in deduplication-intensive scenarios, whereas SSDs and NVMe drives, with their higher speed and lower latency, are better suited in this scenario, particularly because deduplication involves many I/O operations that run more efficiently on SSDs and NVMe drives.

The effect of deduplication on the service life of SSDs and NVMe drives is another important aspect. Because the number of write cycles is limited for these storage types, the frequent write activity associated with deduplication could potentially curtail the service life of these devices. You need to take this into account when planning the storage infrastructure and its maintenance cycles.

In environments with a combination of storage types, it might make sense to store deduplicated data on SSDs or NVMe drives to take advantage of the higher speed, while storing larger and less frequently accessed data on the less expensive HDDs. In this way, the storage space on the more expensive SSDs and NVMe drives can be better utilized through the efficient use of data deduplication. Regardless of the type of storage used, it is crucial to implement robust backup and recovery strategies. Deduplication can increase the complexity of data recovery, requiring careful planning and regular reviews of the backup strategies.

Deduplication Process

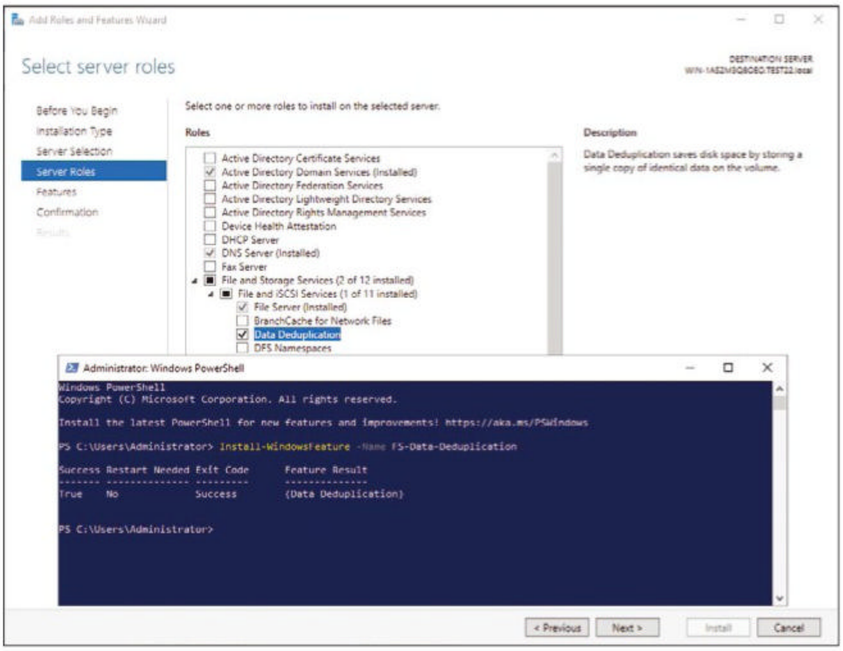
Technically speaking, the deduplication function analyzes the data blocks on a volume and searches for duplicates. As soon as identical data blocks are found, the system only keeps one copy and creates links to this block for each instance of its use. This process is executed by a background service that runs regularly to check for new and modified files.

Deduplication on Windows Server 2022 uses a postprocess approach (i.e., the data is first saved in its original form and then retroactively deduplicated). This approach minimizes the effect on system performance during primary storage operations. For efficient data processing, deduplication relies on a chunking algorithm that breaks the data down into smaller units and then analyzes them individually.

Data integrity is a key aspect of deduplication. Windows Server 2022 uses various mechanisms, including checksums and integrity checks, to ensure that the deduplicated data is not corrupted. Deduplication relies on metadata to manage the original data and the deduplicated copies, requiring additional care for backup and restore operations because the metadata is key to reconstructing the original data correctly.

Installation Two Ways

Data deduplication can be integrated with Server Manager by installing the *Data Deduplication* server role under *File and Storage Services | File and iSCSI Services* (Figure 1). Alternatively, you can run the following command in PowerShell:



Select one or more roles to install on the selected server.

□ Active Directory Certificate Services
☑ Active Directory Domain Services (Installed)
□ Active Directory Federation Services
□ Active Directory Lightweight Directory Services
□ Active Directory Rights Management Services
□ Device Health Attestation
□ DHCP Server
☑ DNS Server (Installed)
□ Fax Server
■ File and Storage Services (2 of 12 installed)
  ■ File and iSCSI Services (1 of 11 installed)
    ☑ File Server (Installed)
    □ BranchCache for Network Files
    ☑ Data Deduplication
    □ DFS Namespaces

Data Deduplication saves disk space by storing a single copy of identical data on the volume.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Administrator> Install-WindowsFeature -Name FS-Data-Deduplication

Success Restart Needed Exit Code      Feature Result
------- -------------- ---------      --------------
True    No             Success        {Data Deduplication}


PS C:\Users\Administrator>
```

Figure 1: Setting up data deduplication in Server Manager and with PowerShell on Windows Server 2019/2022.

```
Install-WindowsFeature ⮐
  -Name FS-Data-Deduplication
```

Installing deduplication does not start the process; it simply imports the required system files. You need to complete the configuration in Server Manager or PowerShell.

Testing Volumes

In the course of installing the server roles for data deduplication, the installation wizard also integrates the `ddpeval.exe` command-line tool. You can use it at the command line to search for duplicate files (Figure 2). Doing so will tell you whether the server role can be meaningfully applied to the individual data carriers on the server. You cannot enable data deduplication on boot drives or use `ddpeval` to check whether data deduplication on boot drives makes sense. The tool resides in the `\Windows\System32` directory and supports both local drives and network shares. The syntax of the tool is `ddpeval <Volume:>`, as in:

```
ddpeval E:\
ddpeval \\nas\data
```

The `ddpeval` tool itself does not clean up the files; it simply tells you whether or not data deduplication makes sense for the drive in question and offers a preview of possible savings through data deduplication without modifying the data. For a more targeted analysis of a specific directory, you need to modify the command as follows:

```
ddpeval.exe D:\Data\Projects
```

The output from `ddpeval` contains details of the total size of the analyzed data, the estimated size after deduplication, and the savings as a percentage. This information is crucial for making an informed decision on implementing data deduplication. In particular, the tool helps you evaluate the potential benefits of deduplication and decide which volumes or directories are best suited for deduplication.

The following command lets you save the results:

```
DDPEval.exe d:\wsus /v /o:C:\temp\dedup.txt
```

This syntax gives you a comprehensive report on the potential storage space savings that you can achieve by introducing data deduplication.

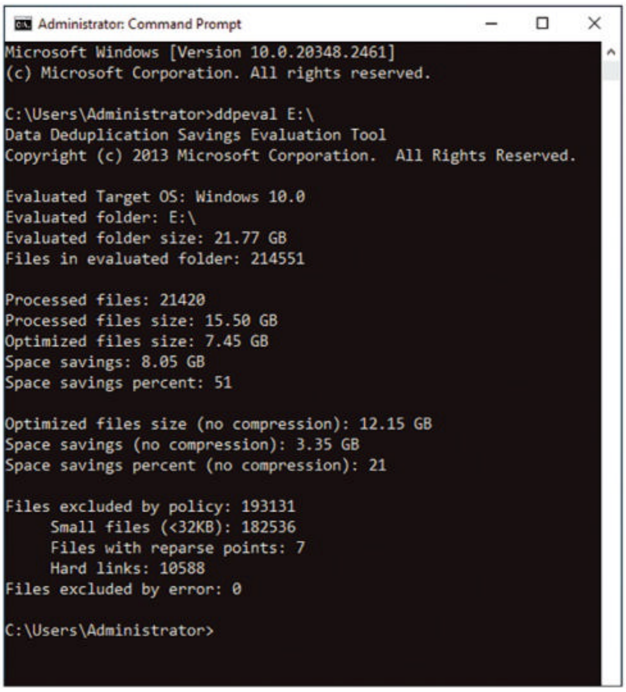
Data Deduplication for Volumes

After installing the server role for data deduplication and testing the individual drives, the next step is to enable the feature for the target drives on the target server. To do this, you can either use the Server Manager and enable the function by selecting *File and Storage Services* | *Volumes* followed by *Configure Data Deduplication* in the target volume's context menu, or you can use PowerShell if you prefer. I will be looking at both options later. Selecting the option in Server Manager pops up a window where you can configure all the settings required for the target volume.

Start by selecting the server type and the data to be deduplicated. *General purpose file server*, *Virtual Desktop Infrastructure (VDI) server*, and *Virtualized Backup Server* are available as options. Specify the number of days to wait before deduplicating duplicate files (Figure 3). A period of three days is preconfigured by default. You can also exclude individual file types, individual files, or entire folders from deduplication. Click the *Set Deduplication Schedule* button to set up in detail when you want the background service to clean up the server. You will generally want to check the *Enable background optimization* option, which means that the deduplication service will run in the background and generate as little load as possible on the server. Windows can even stop the service if required. In the window, you can also define two additional schedules for days on which deduplication will run with normal priority at specific times. Of course, you will want to select times when the server is not very busy. As a general rule, you should avoid other activities taking place on the server at the same time as deduplication, including maintenance, data backups, and malware scans.

Deduplication of VDI Servers

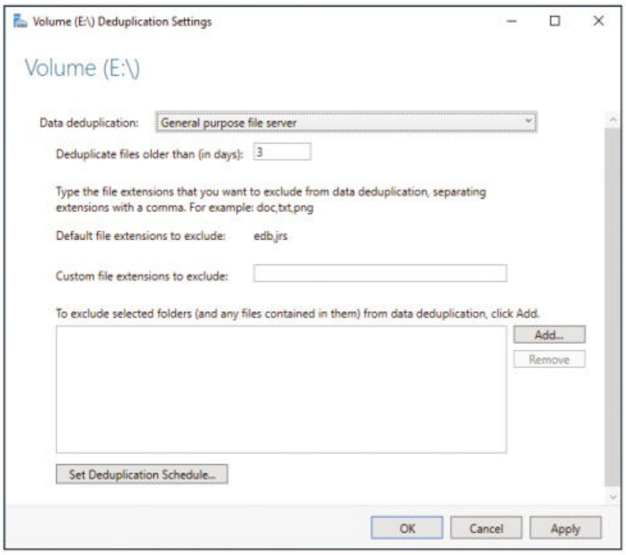
Data deduplication in VDI environments offers substantial benefits, but some key aspects differ from deduplication on conventional file servers. VDI scenarios often have many desktop instances with similar or identical data, and deduplication can achieve significant storage space savings by eliminating redundant data across multiple virtual desktops.



```
Microsoft Windows [Version 10.0.20348.2461]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ddpeval E:\
Data Deduplication Savings Evaluation Tool
Copyright (c) 2013 Microsoft Corporation.  All Rights Reserved.

Evaluated Target OS: Windows 10.0
Evaluated folder: E:\
Evaluated folder size: 21.77 GB
Files in evaluated folder: 214551

Processed files: 21420
Processed files size: 15.50 GB
Optimized files size: 7.45 GB
Space savings: 8.05 GB
Space savings percent: 51

Optimized files size (no compression): 12.15 GB
Space savings (no compression): 3.35 GB
Space savings percent (no compression): 21

Files excluded by policy: 193131
     Small files (<32KB): 182536
     Files with reparse points: 7
     Hard links: 10588
Files excluded by error: 0

C:\Users\Administrator>
```

Figure 2: Use the command line to discover whether data deduplication makes sense for individual drives.

This process does not just reduce the storage capacity you need, it also improves performance, because less physical storage space is required to store and read the data.



Volume (E:\) Deduplication Settings

Volume (E:\)

Data deduplication: General purpose file server

Deduplicate files older than (in days): 3

Type the file extensions that you want to exclude from data deduplication, separating extensions with a comma. For example: doc,txt,png

Default file extensions to exclude: edb,jrs

Custom file extensions to exclude:

To exclude selected folders (and any files contained in them) from data deduplication, click Add.

Figure 3: Setting up data deduplication for individual drives in Server Manager.

One significant difference from deduplication on file servers relates to the type of data stored. Whereas file servers usually store a variety of file types and data structures, the files in a VDI environment are often more homogeneous, because many virtual machines use similar operating systems and applications. This homogeneity increases the potential for deduplication because more redundant data exists. Additionally, deduplication in VDI environments often requires a customized configuration to meet the specific requirements of these environments. For example, it can be important to configure deduplication such that it does not affect performance at peak times; after all, response times and availability are critical factors in VDI environments. Another difference lies in maintenance and administration. VDI environments can be dynamic by nature, with frequent changes to virtual desktops, requiring regular reviews and adjustments of the deduplication settings. In contrast, the content on the file servers is often more static, which means the deduplication settings do not need to be modified as frequently.

PowerShell

To use PowerShell to control data deduplication on Windows Server 2022, the following commands enable data

deduplication for a target volume and configure the settings:

```
Enable-DedupVolume -Volume F:
Enable-DedupVolume -Volume d: ⮐
                   -UsageType Default
```

You can also manage this process with *General purpose file server* in Server Manager, and you can immediately start deduplication with the command

```
Start-DedupJob -Volume <drive letter> ⮐
               -Type Optimization
```

`Set-DedupSchedule` modifies the configuration of the deduplication parameters, such as the schedule for garbage collection and optimization:

```
Set-DedupSchedule ⮐
  -Name "DailyOptimization" ⮐
  -Type Optimization ⮐
  -Start 01:00 ⮐
  -DurationHours 3
```

You can use the following PowerShell command to discover the scheduled tasks:

```
Get-ScheduledTask ⮐
  -TaskPath \Microsoft\Windows\⮐
           Deduplication\
```

`Get-DedupStatus` lets you monitor the deduplication rate and savings achieved, whereas

```
Start-DedupJob -Volume "D:" -Type Scrubbing
```

checks the integrity of the deduplicated data. These commands give you comprehensive options for controlling and monitoring data deduplication without Server Manager. If you want to wait for the deduplication response, type

```
Start-DedupJob <drive letter> ⮐
  -Type Optimization -Wait
```

You can also display the current status of a job and retrieve further information by typing

```
Get-DedupJob
Get-DedupVolume
```

For more detailed information, you can redirect the output to the `Format-List` cmdlet (e.g., `Get-DedupVolume | fl`). Careful monitoring of deduplication success is also important. You can create reports with commands such as

```
Get-DedupVolume -Volume "D:" | ⮐
  Select-Object SavingsRate,⮐
                OptimizedFilesCount
```

to output deduplication success metrics and make adjustments, if necessary. PowerShell also lets you configure the various deduplication options. For example, you can adjust the minimum file size for deduplication to improve efficiency:

```
Set-DedupVolume -Volume "D:" ⮐
  -MinimumFileSize 128KB
```

You can also disable additional compression with the `NoCompress` parameter if the data is already compressed. Certain file types can be excluded from deduplication to optimize performance for these files, which is not only possible in Server Manager, but also in PowerShell:

```
Set-DedupVolume -Volume "D:" ⮐
  -ExcludeFileType "log","tmp"
```

If you want to disable data deduplication for a drive again, you can use Server Manager from the same window as for enabling deduplication. To do so, set the *Disabled* option in *Configure Data Deduplication*. If you want to use PowerShell instead, run the command

```
Disable-DedupVolume -Volume F:
```

In some circumstances, you might need to restore deduplicated volumes, for which you can use

```
Start-DedupJob -Volume "D:" ⮐
  -Type Unoptimization
```

This kind of flexibility is particularly useful in complex IT environments. Windows Server 2022 offers special optimization options for custom applications such as VDI environments. Adjusting the settings with commands such as

```
Set-DedupVolume -Volume "D:" ⮐
  -OptimizeInUseFiles ⮐
  -OptimizePartialFiles
```

maximizes deduplication performance in these environments. It is also advisable to carry out regular checks and maintenance to ensure that your data deduplication setup is running efficiently and without interruptions. When planning deduplication tasks, you also need to take the server load into account. Scheduling deduplication tasks outside of peak times helps you minimize the server load and optimize the overall performance.

Conclusions

Data deduplication in Windows Server 2022 is a powerful tool that can significantly improve the efficiency and management of server storage resources. One of the main advantages is the significant reduction in storage space requirements by eliminating redundant data on the server, which in turn drives cost savings and enables more efficient storage management, especially in environments with large volumes of data. Network bandwidth is also conserved by reducing the data transfer volumes for backup and replication.

On the downside, deduplication can generate CPU and memory load, potentially causing performance issues in server environments that are already heavily utilized. Also, some file types do not lend themselves to deduplication – specifically, compressed data or data prone to rapid change, such as databases or real-time logfiles. ■

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].

Linux Magazine Subscription

Print and digital options
12 issues per year

LINUX MAGAZINE
Artisan Keycaps: Control the narrative with the ultimate Ctrl key
Virtual Memory
Explore the secrets of the swap space
Zint: Create your own barcodes and QR codes
D-Bus
Learn how Linux applications ... ther
Pivot Tables: Scale up your spreadsheet skills
Jellyfin: Movie night with a Rasp Pi
10 APPLAUDABLE APPS FOR YOUR LINUX TOOLKIT
Snapshot: Create a pool game simulation in Go
AI Tools
Practical apps with intelligence built in
Nobara: Fedora fork for gamers
OpenWRT: Use a router OS on an old Rasp Pi
FreeRTOS: Realtime OS for microcontrollers
Shinobi: Set up a surveillance camera network
LosslessCut: Cut and paste multimedia clips – without losing quality
7 SNAPPY FOSS FINDS!
WWW.LINUX-MAGAZINE.COM

SUBSCRIBE
shop.linuxnewmedia.com

Expand your Linux skills:

- In-depth articles on trending topics, including Bitcoin, ransomware, cloud computing, and more!
- How-tos and tutorials on useful tools that will save you time and protect your data
- Troubleshooting and optimization tips
- Insightful news on crucial developments in the world of open source
- Cool projects for Raspberry Pi, Arduino, and other maker-board systems

Go farther and do more with Linux, subscribe today and never miss another issue!

Follow us

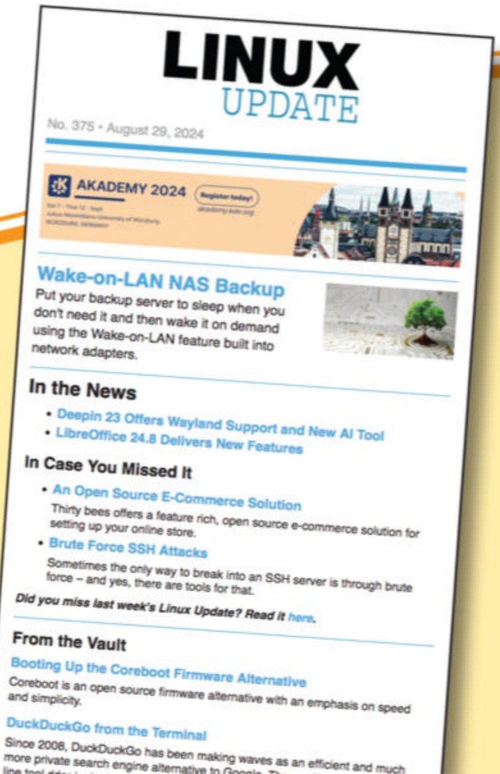
@linux_pro
Linux Magazine
@linuxpromagazine
@linuxmagazine

Need more Linux?

Subscribe free to Linux Update

Our free Linux Update newsletter delivers insightful articles and tech tips to your inbox every week.

https://bit.ly/Linux-Update



LINUX UPDATE

No. 375 • August 29, 2024

Wake-on-LAN NAS Backup

Put your backup server to sleep when you don't need it and then wake it on demand using the Wake-on-LAN feature built into network adapters.

In the News

- Deepin 23 Offers Wayland Support and New AI Tool
- LibreOffice 24.8 Delivers New Features

In Case You Missed It

- An Open Source E-Commerce Solution

  Thirty bees offers a feature rich, open source e-commerce solution for setting up your online store.
- Brute Force SSH Attacks

  Sometimes the only way to break into an SSH server is through brute force – and yes, there are tools for that.

Did you miss last week's Linux Update? Read it here.

From the Vault

Booting Up the Coreboot Firmware Alternative

Coreboot is an open source firmware alternative with an emphasis on speed and simplicity.

DuckDuckGo from the Terminal

Since 2008, DuckDuckGo has been making waves as an efficient and much more private search engine alternative to Google.

Integrate remote cloud storage

Easy Access

You don't need native clients for every single service just to back up or synchronize your data in the cloud; Rclone helps you handle these tasks for multiple cloud accounts at the command line or in a graphical front end. By Erik Bärwaldt

Often, users fail to back up their data regularly and synchronize backups. Of course, any data you store locally should also be stored on a remote mass storage device. If the local server or network-attached storage (NAS) is stolen or destroyed by natural events (e.g., floods, fires), the backups they store will also be lost. Because most companies and many private individuals now use cloud storage, it makes sense to transfer backups to the cloud, and Rclone [1] is the ideal solution.

Strategy

The standard command-line tool for backing up data on remote computers on the intranet is Rsync, but Rclone is far better suited for backing up data to the cloud. The learning curve is manageable because the tool is based on Rsync syntax (Table 1). Rclone can be easily installed on most popular distributions with built-in package managers.

The latest versions are also available for download on the project's website for other operating systems, including various BSD derivatives and Solaris. Rclone is controlled from the command line, but the developers have also provided a graphical front end [2], although it is still in an experimental phase, so you might want to avoid using it in production environments.

The application was largely developed in the Go programming language and is available under an MIT license. Rclone can back up and synchronize entire databases, as well as individual files and directories, in the cloud, and you can encrypt and synchronize backups between different cloud services. Rclone supports more than 70 cloud providers, as well as individual systems such as on-premises SFTP servers.

For each supported provider, the developers have detailed individual configuration instructions [3] so that even beginners can set up the tool for use with an existing cloud account. Rclone

Table 1: **Rsync vs. Rclone**

Feature	Rsync	Rclone
License	GPLv3	MIT
Intended use	Data backup between two computers or servers	Data backup to the cloud, treating cloud storage like local drives
Backup type	Unidirectional	Unidirectional
Threads	Single-threaded	Multithreaded
Transfer method	Single file blocks	Complete files
Integrity check	Yes	Yes
Support for cloud services	No	Yes
Mount target drives	No	Yes

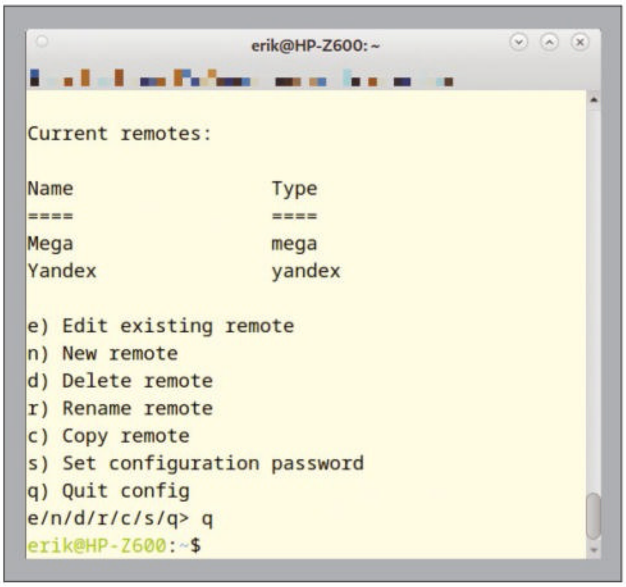
Lead Image © rawintanpin, 123RF.com

can also use a cron job to synchronize data automatically, making sure your backups are always up to date.

Installation

The easiest way to install Rclone is to use the software repositories provided by your chosen distribution. If you only find an older version there, simply retrieve Rclone from the project's GitHub page. You can also use the script to install it on your system, for which you need `curl`. To install `curl`, integrate Rclone into your system, and start the basic configuration in a wizard, enter:

```
sudo apt install curl
$ curl https://rclone.org/install.sh | ⮐
  sudo bash
$ rclone config
```



```
Current remotes:

Name                 Type
====                 ====
Mega                 mega
Yandex               yandex

e) Edit existing remote
n) New remote
d) Delete remote
r) Rename remote
c) Copy remote
s) Set configuration password
q) Quit config
e/n/d/r/c/s/q> q
erik@HP-Z600:~$
```

Figure 1: The configuration wizard runs in a terminal window.

The first step in the wizard asks for the *remote* (i.e., the target system). Pressing *n* opens the dialog for configuring a new cloud service. You will see a list of storage services, each of which has its own number. Simply enter the number that matches your provider.

The script then requests the credentials for the service, which can be location information or authentication data. The wizard does not display the password, but you do need to type it in a second time to confirm.

In the next step, the access credentials you entered are displayed and the wizard prompts you to confirm. If you make a mistake, you can edit the entry by pressing *e* or delete it completely by pressing *d*. If the data is correct, press *y*. The wizard now displays the external service along with a selection of options, including an option for setting up another service (Figure 1). When done, press *q* to quit the configuration wizard.

Alternatively, you can install Rclone by downloading the package from the website and using your distribution's software manager to install. RPM and DEB packages for 32- and 64-bit Linux systems are available on the website, along with packages for various versions of the ARM architecture. A ZIP archive [4] for Linux is also available that contains a generic binary package of the software. Simply unzip the file in a directory of your choice, change to that directory, and call up the configuration wizard as described above to configure the basic settings. The software is then ready for use.

If you already have an older version of the program in place, you can update it by running the `rclone selfupdate` command. If required, you can use the `rclone version` command to query the currently installed version (Figure 2).

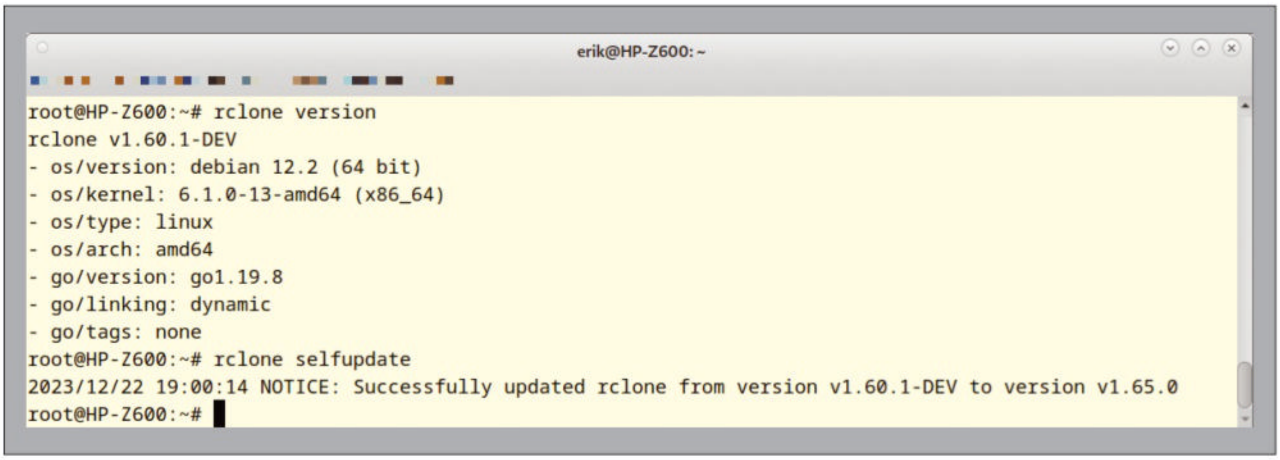
For an overview of the software's numerous functions and parameters, type `rclone -h` at the prompt. Detailed help is available for every option, which you can access with the

```
rclone <option> --help
```

command. You can also combine parameters.

Network Drive

To set up cloud storage for this scenario, begin by creating a new directory in your personal folder before mounting your cloud storage on your system like a conventional network drive:



```
root@HP-Z600:~# rclone version
rclone v1.60.1-DEV
- os/version: debian 12.2 (64 bit)
- os/kernel: 6.1.0-13-amd64 (x86_64)
- os/type: linux
- os/arch: amd64
- go/version: go1.19.8
- go/linking: dynamic
- go/tags: none
root@HP-Z600:~# rclone selfupdate
2023/12/22 19:00:14 NOTICE: Successfully updated rclone from version v1.60.1-DEV to version v1.65.0
root@HP-Z600:~# █
```

Figure 2: Update the program with just a few commands.

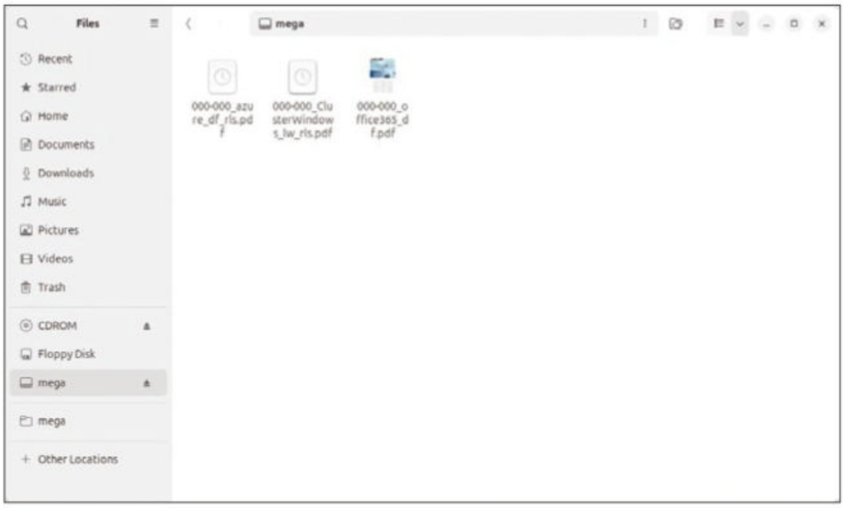
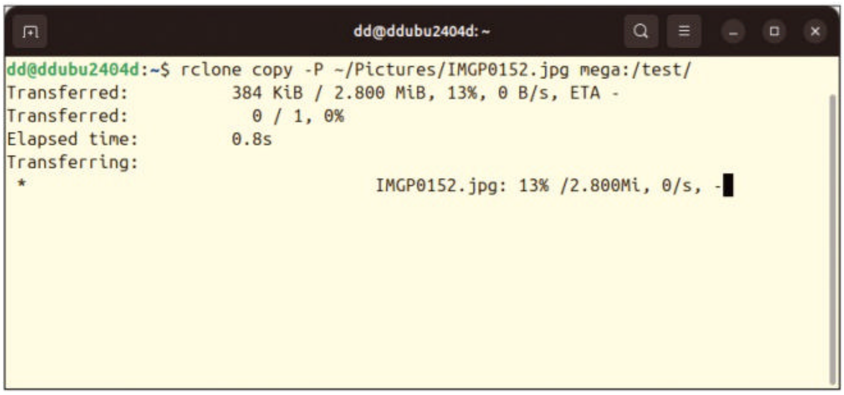


Figure 3: Rclone mounts cloud storage on your system like a network drive.



```
dd@ddubu2404d:~$ rclone copy -P ~/Pictures/IMGP0152.jpg mega:/test/
Transferred:          384 KiB / 2.800 MiB, 13%, 0 B/s, ETA -
Transferred:            0 / 1, 0%
Elapsed time:         0.8s
Transferring:
 *                                IMGP0152.jpg: 13% /2.800Mi, 0/s, -
```

Figure 4: Copying individual files from the command line.

```
$ rclone ↩
  --vfs-cache-mode writes mount <service> ↩
  ~/<folder>
```

The `<service>` parameter refers to the cloud service you defined in the Rclone configuration phase, and `<folder>` refers to the newly created target folder. Please note that the spelling of the service name when entering the command must be exactly the same as when configuring Rclone. Depending on your Internet access speed and the volume of data to be transferred, the complete folder structure of the cloud storage will appear in the target folder on your system after a wait (Figure 3). You now have full access to all content stored in the cloud, provided you also granted all rights for this service in the Rclone configuration.

In your desktop's file manager, you can then use the cloud directory and its subfolders in the same way as local directories. Copying files from the cloud to the local system is also a painless experience, but note that content, particularly larger documents such as PDF files, will take far longer to load than local files.

Also note that you do need to re-enter the command for mounting your cloud storage as a network drive every time your restart the system. It makes sense to create a script to integrate cloud storage automatically after a restart.

Copying Data

The `copy` subcommand tells Rclone to copy individual files to the cloud. For example, in the command

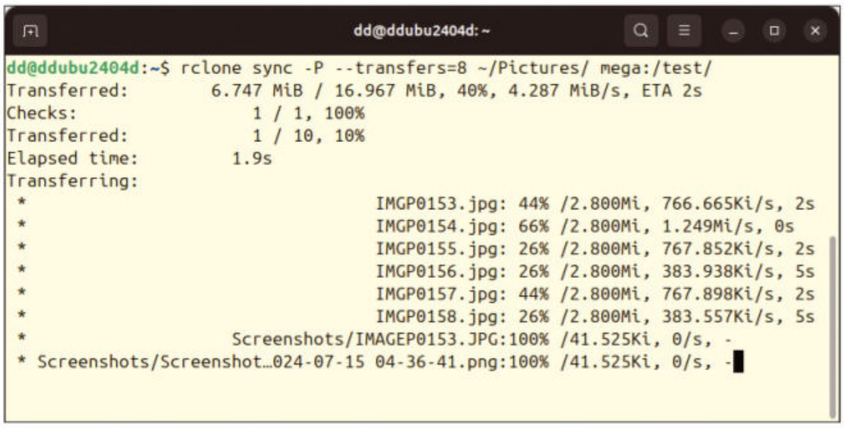
```
$ rclone copy <Source>/<File> ↩
             <Target>:/<Path>/
```

`<Target>` refers to the cloud service defined in the Rclone configuration wizard. To display progress during the `copy`, you can also add the `-P` parameter (Figure 4). The application then displays both the transfer history and the time required, once the data transfer has completed. Note that Rclone overwrites existing files of the same name in the target directory without a warning prompt.

The application supports numerous flags (e.g., to avoid having to write the entire database from the source folder to the cloud every time you make a regular copy). To transfer just modified or newly added files, you can use the `--max-age <age>` flag to back up only the files that were modified during the specified time period. You can specify the time format in any increment from milliseconds to years. The flags are described in detail in the software documentation.

Synchronizing Data

When synchronizing data between two drives, Rclone has an option to



```
dd@ddubu2404d:~$ rclone sync -P --transfers=8 ~/Pictures/ mega:/test/
Transferred:        6.747 MiB / 16.967 MiB, 40%, 4.287 MiB/s, ETA 2s
Checks:                 1 / 1, 100%
Transferred:            1 / 10, 10%
Elapsed time:         1.9s
Transferring:
 *                                IMGP0153.jpg: 44% /2.800Mi, 766.665Ki/s, 2s
 *                                IMGP0154.jpg: 66% /2.800Mi, 1.249Mi/s, 0s
 *                                IMGP0155.jpg: 26% /2.800Mi, 767.852Ki/s, 2s
 *                                IMGP0156.jpg: 26% /2.800Mi, 383.938Ki/s, 5s
 *                                IMGP0157.jpg: 44% /2.800Mi, 767.898Ki/s, 2s
 *                                IMGP0158.jpg: 26% /2.800Mi, 383.557Ki/s, 5s
 *                    Screenshots/IMAGEP0153.JPG:100% /41.525Ki, 0/s, -
 * Screenshots/Screenshot…024-07-15 04-36-41.png:100% /41.525Ki, 0/s, -
```

Figure 5: Rclone synchronizes large databases by transferring several files in parallel.

delete data that no longer exists in the source folder from the target drive.

The sync subcommand synchronizes locally stored content with the cloud; you'll only need the name of the root directory, not file names:

```
$ rclone sync ↩
  -P /<Source>/ <Target>:/<Path>/
```

Rclone copies the data to the cloud and keeps the folder structure. The -P parameter displays a progress bar. Rclone then displays the names of the individual files that it is currently synchronizing with the cloud. By default, the tool transfers up to four files simultaneously. If required, you can set a higher number of simultaneous transfers with the --transfers=<number> parameter (Figure 5).

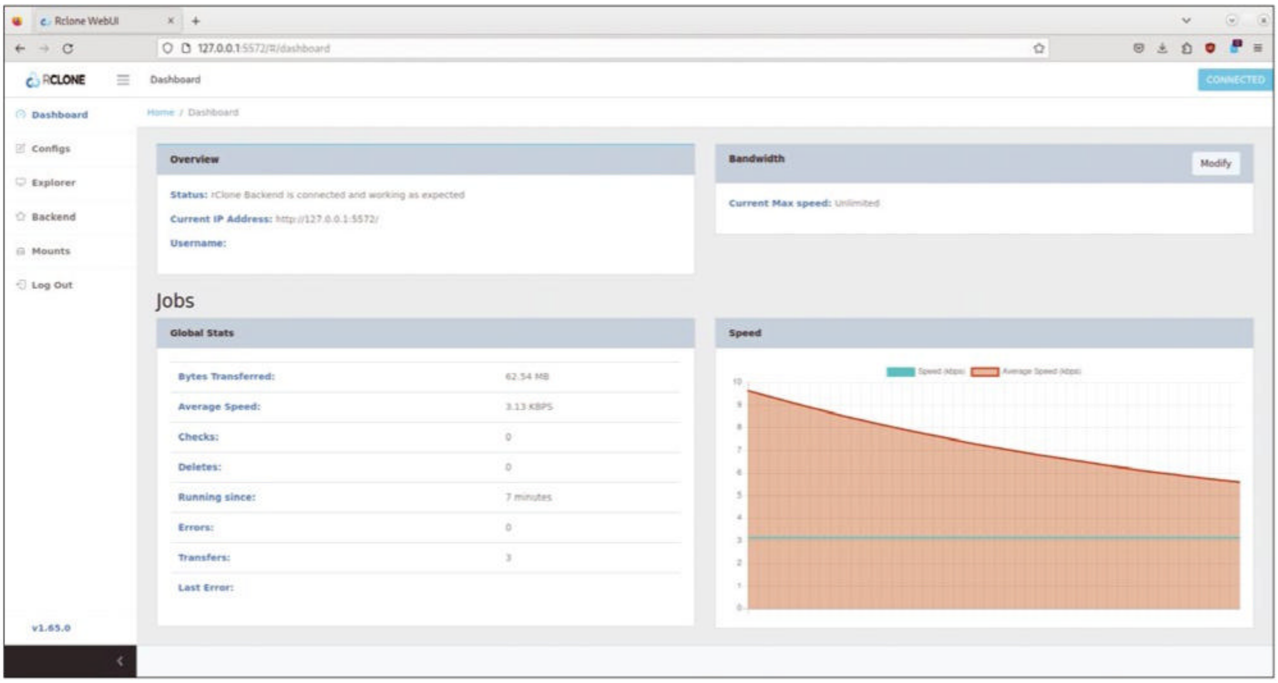
Graphical Front End

The developers of Rclone provide a graphical, web-based front end for the application. The command

```
rclone rcd --rc-web-gui
```

retrieves the package from the project's GitHub page and sets it up. After the install, the graphical dashboard (Figure 6) automatically appears in your local web browser. You can call it up again later at any time by typing *http://127.0.0.1:5572/* in the address bar.

The front end gives you direct access to the content of the target cloud account. You can use the *Explorer* file manager option in the left control bar to browse the content, navigate between the individual directory trees, and download or share files from the

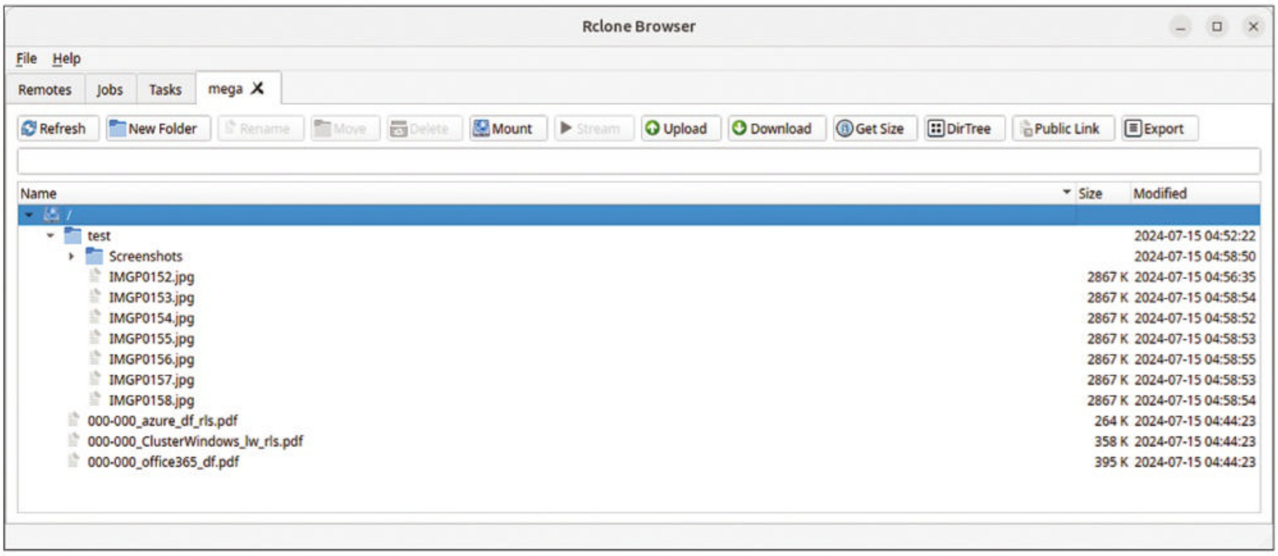


Status: rClone Backend is connected and working as expected
Current IP Address: http://127.0.0.1:5572/
Username:

Current Max speed: Unlimited

Bytes Transferred:	62.54 MB
Average Speed:	3.13 KBPS
Checks:	0
Deletes:	0
Running since:	7 minutes
Errors:	0
Transfers:	3
Last Error:	

Figure 6: The Rclone graphical user interface looks very professional.



Name	Size	Modified
/		
test		2024-07-15 04:52:22
Screenshots		2024-07-15 04:58:50
IMGP0152.jpg	2867 K	2024-07-15 04:56:35
IMGP0153.jpg	2867 K	2024-07-15 04:58:54
IMGP0154.jpg	2867 K	2024-07-15 04:58:52
IMGP0155.jpg	2867 K	2024-07-15 04:58:53
IMGP0156.jpg	2867 K	2024-07-15 04:58:55
IMGP0157.jpg	2867 K	2024-07-15 04:58:53
IMGP0158.jpg	2867 K	2024-07-15 04:58:54
000-000_azure_df_rls.pdf	264 K	2024-07-15 04:44:23
000-000_ClusterWindows_lw_rls.pdf	358 K	2024-07-15 04:44:23
000-000_office365_df.pdf	395 K	2024-07-15 04:44:23

Figure 7: Rclone Browser makes using the software child's play.

cloud. You can also create new directories in the cloud or upload content with the buttons displayed top right in the window. A search function can help you search for specific file formats, and the *Configs* option lets you integrate new cloud connections in the graphical user interface or update or delete existing connections.

Rclone Browser

Rclone Browser [5] offers a convenient graphical user interface for Rclone. You can download this application from the project's GitHub page as an app image for 32- and 64-bit PCs and the Raspberry Pi. Once execute rights have been granted,

```
chmod +x <image>
```

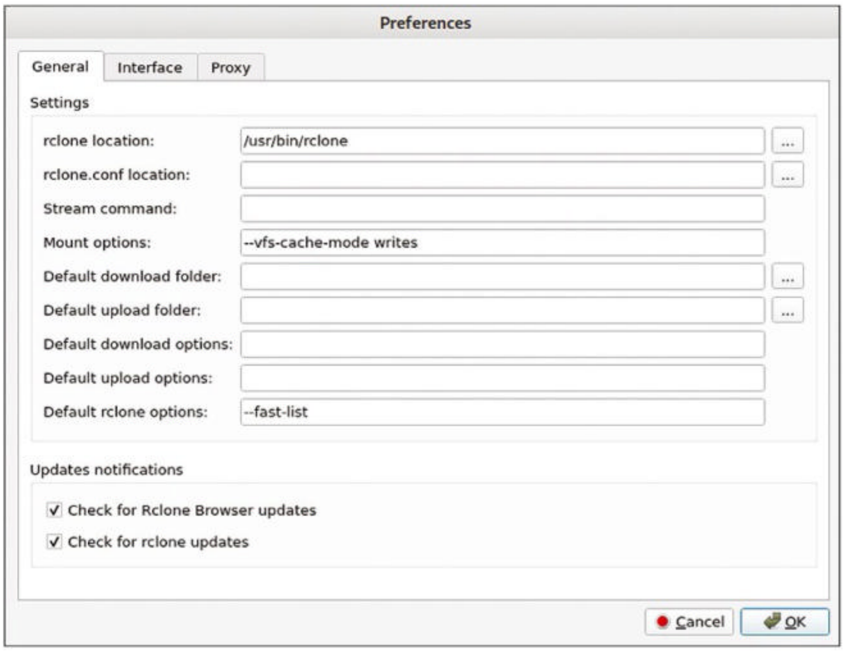
the images can be used independent of the package manager on any Linux distribution without the need to install. Calling the program opens a standard program window with a menubar, a tab bar, and a buttonbar. Start by going to the *Remotes* tab and press the *Refresh* button below to the left to display the cloud services configured in Rclone. Selecting a service opens it and displays the content in a tree view in the new tab created for the service (Figure 7).

You can navigate freely in the tree view. To edit content, click on the desired directory or file, which enables the function buttons for the file or directory in the toolbar above the tree view. You can use them to upload, download, rename, and delete files. Dialog boxes let you tweak the settings for uploading or downloading. Regardless of the content currently selected, you can create new folders and rename or move existing ones at any time. The application can also display the folder size and the number of files it contains. You can view existing control actions such as cron jobs in the *Jobs* and *Tasks* tabs, provided you used Rclone to define them.

Instead of the toolbar, you can use the context menus with a right-click on a file or directory in the tree view to work with your cloud content.

Customization

Pressing *File* | *Preferences* brings up the Preferences dialog (Figure 8).



Preferences

Settings

rclone location: /usr/bin/rclone

rclone.conf location:

Stream command:

Mount options: --vfs-cache-mode writes

Default download folder:

Default upload folder:

Default download options:

Default upload options:

Default rclone options: --fast-list

Updates notifications

☑ Check for Rclone Browser updates

☑ Check for rclone updates

Figure 8: The Preferences dialog has parameters and flags you use to customize your Rclone Browser settings.

Under the *General* tab, you can specify the Browser start parameters that Rclone uses when opening connections to the cloud services. You can also predefine upload and download folders here.

In the *Interface* tab, you can specify various options for the appearance and behavior of the application. The *Proxy* tab offers a range of configuration options for a proxy server. The settings made here affect all activated cloud connections.

Conclusions

Rclone is the perfect counterpart for admins who use the Rsync tool on local networks by helping them integrate cloud services into their data backup strategy. With its various operating modes, it not only supports typical deployment options, but also lets you use multiple cloud services simultaneously without having to install native clients up front. Various graphical front ends add more convenience for the tasks at hand. Cloud storage can be seamlessly mounted on the local system as a network drive, but given the diverse feature set, make sure you plan time to familiarize yourself with the software. If you want to use Rclone at work, go the extra mile and customize the application to precisely meet your needs. ■

Info

[1] Rclone: [https://rclone.org]

[2] Graphical front end: [https://rclone.org/gui/]

[3] Documentation: [https://rclone.org/docs/]

[4] Generic binary: [https://github.com/rclone/rclone/releases/tag/v1.65.0]

[5] Rclone Browser download: [https://kapitainsky.github.io/RcloneBrowser/]

Author

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

Hone Your Skills – with – Special Issues!

JOIN THE LINUX REVOLUTION!
GETTING STARTED WITH LINUX
FIND THE PERFECT TOOL FOR THE TASK
Cool Linux Hacks
Tricks and shortcuts for Linux geeks
114 HACKS
301 BEST BASH COMMANDS
LINUX SHELL HANDBOOK
SUPERCHARGE YOUR LINUX SKILLS
Dive deep into the world's greatest free office suite
LibreOffice Expert

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

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

Available in print or digital format

RASPBERRY PI GEEK
BUILD A RASP PI RADIO!
MakerSpace #02
RASP PI TRICKS
FEED YOUR FISH WHILE YOU'RE AWAY
RASP PI TRICKS
TRY ARTIFICIAL INTELLIGENCE ON A PI
MakerSpace #04
HANDS-ON PROJECTS FOR MAKERS

background Image © roystudio, 123RF.com

Check out the full library!

shop.linuxnewmedia.com

ADMIN Network & Security

NEWSSTAND

Order online:
https://bit.ly/ADMIN-library

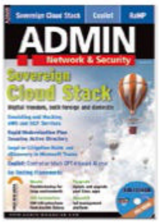
ADMIN is your source for technical solutions to real-world problems. Every issue is packed with practical articles on the topics you need, such as: security, cloud computing, DevOps, HPC, storage, and more! Explore our full catalog of back issues for specific topics or to complete your collection.

#82 – July/August 2024

Sovereign Cloud Stack

SCS liberates your data centers from monopolistic operations and companies beholden to out-country laws and regulations.

On the DVD: Kali Linux 2024.2

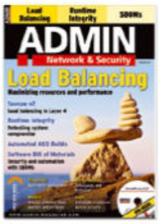


#81 – May/June 2024

Load Balancing

Load balancing on heavily frequented networks improves performance, availability, security, scalability, and the ability to handle peak loads.

On the DVD: SystemRescue 11.01

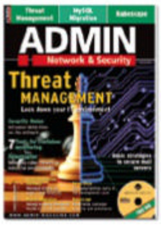


#80 – March/April 2024

Threat Management

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

On the DVD: openSUSE Leap 15.5

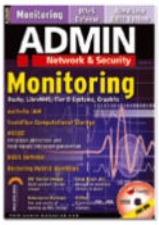


#79 – January/February 2024

Monitoring

This issue takes a deep dive into monitoring solutions for your IT infrastructure, including Dashy, LibreNMS, Tier 0 systems, and Graphite.

On the DVD: FreeBSD 14.0

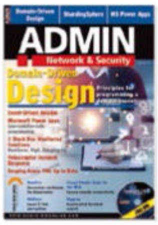


#78 – November/December 2023

Domain-Driven Design

Business experts and developers collaborate to define domain models and business patterns that guide software development.

On the DVD: Fedora Server 39

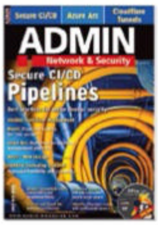


#77 – September/October 2023

Secure CI/CD Pipelines

DevSecOps blends security into every step of the software development cycle.

On the DVD: IPFire 2.27



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
Erik Bärwaldt	90
Norbert Deuschle	16
Markus Feilner	32
Marcin Gastol	76
Ken Hess	3
Thomas Joos	62, 84
Christian Knermann	26
Martin Kuppinger	40
Oliver Kurowski	44
Martin Gerhard Loschwitz	56
Thomas Reuß	66
Ariane Rüdiger	20
Thorsten Scherf	38, 72
Andreas Stolzenberger	10, 52
Matthias Wübbeling	70

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 Pettle, Aubrey Vaughn

Layout
Dena Friesen, Lori White

Cover Design
Dena Friesen, Illustration based on graphics by rouslan, 123RF.com

Advertising
Jessica Pryor, jpryor@linuxnewmedia.com

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.

Printed in Nuremberg, Germany by Kolibri Druck.

Distributed by Seymour Distribution Ltd, United Kingdom

ADMIN (Print ISSN: 2045-0702, Online ISSN: 2831-9583, USPS No: 347-931) is published bimonthly by Linux New Media USA, LLC, and distributed in the USA by Asendia USA, 701 Ashland Ave, Folcroft PA. September/October 2024. Application to Mail at Periodicals Postage Prices is pending at Philadelphia, PA and additional mailing offices. POSTMASTER: send address changes to Linux Magazine, 4840 Bob Billings Parkway, Ste 104, Lawrence, KS 66049, USA.

Represented in Europe and other territories by: Sparkhaus Media GmbH, Bialasstr. 1a, 85625 Glonn, Germany.

ADMIN 84

Available Starting December 6

Our next issue will be packed with all the great content you expect from *ADMIN*. Here are a few of the upcoming articles:

- NoSQL and Vector Databases
- Key-Value Stores vs. Relational Databases
- NVMe-oF as iSCSI Replacement
- Migrating Git Repositories
- And much more!

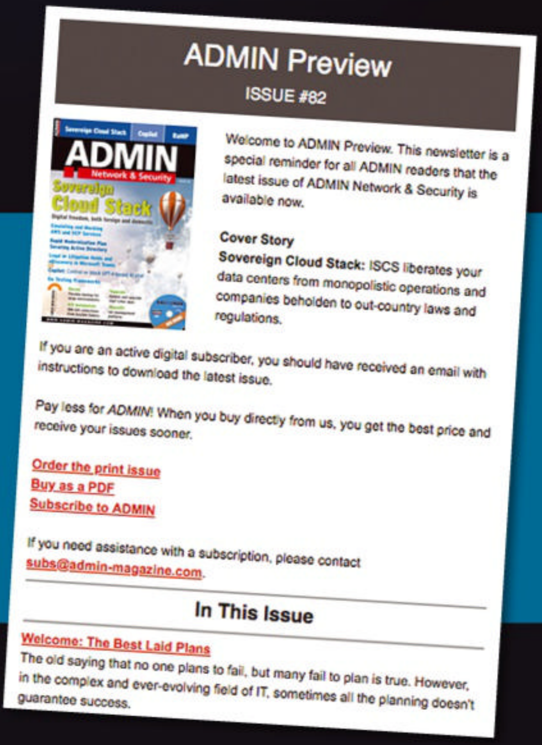
Please note: Articles could change before the next issue.

BE THE FIRST TO SEE WHAT'S NEXT

Subscribe free to the *ADMIN* Preview newsletter and get a sneak peek at every article included in the next issue of *ADMIN*.

Sign up today at https://bit.ly/admin-preview

Image © artnovielysa, 123RF.com



ADMIN Preview

ISSUE #82

Welcome to ADMIN Preview. This newsletter is a special reminder for all ADMIN readers that the latest issue of ADMIN Network & Security is available now.

Cover Story

Sovereign Cloud Stack: ISCS liberates your data centers from monopolistic operations and companies beholden to out-country laws and regulations.

If you are an active digital subscriber, you should have received an email with instructions to download the latest issue.

Pay less for *ADMIN*! When you buy directly from us, you get the best price and receive your issues sooner.

Order the print issue
Buy as a PDF
Subscribe to ADMIN

If you need assistance with a subscription, please contact subs@admin-magazine.com.

In This Issue

Welcome: The Best Laid Plans

The old saying that no one plans to fail, but many fail to plan is true. However, in the complex and ever-evolving field of IT, sometimes all the planning doesn't guarantee success.

CHECK OUT
HETZNER CLOUD NOW!