

# GWDG NACHRICHTEN 09|22

Drucken und Scannen  
im neuen Göttinger  
Rechenzentrum

.....  
Neue DNS-Infrastruktur

.....  
Task Dependent Workflow  
with Slurm

.....  
KubeCon and  
OpenInfra 2022

ZEITSCHRIFT FÜR DIE KUNDEN DER GWDG

## Domain Name Server



DNS

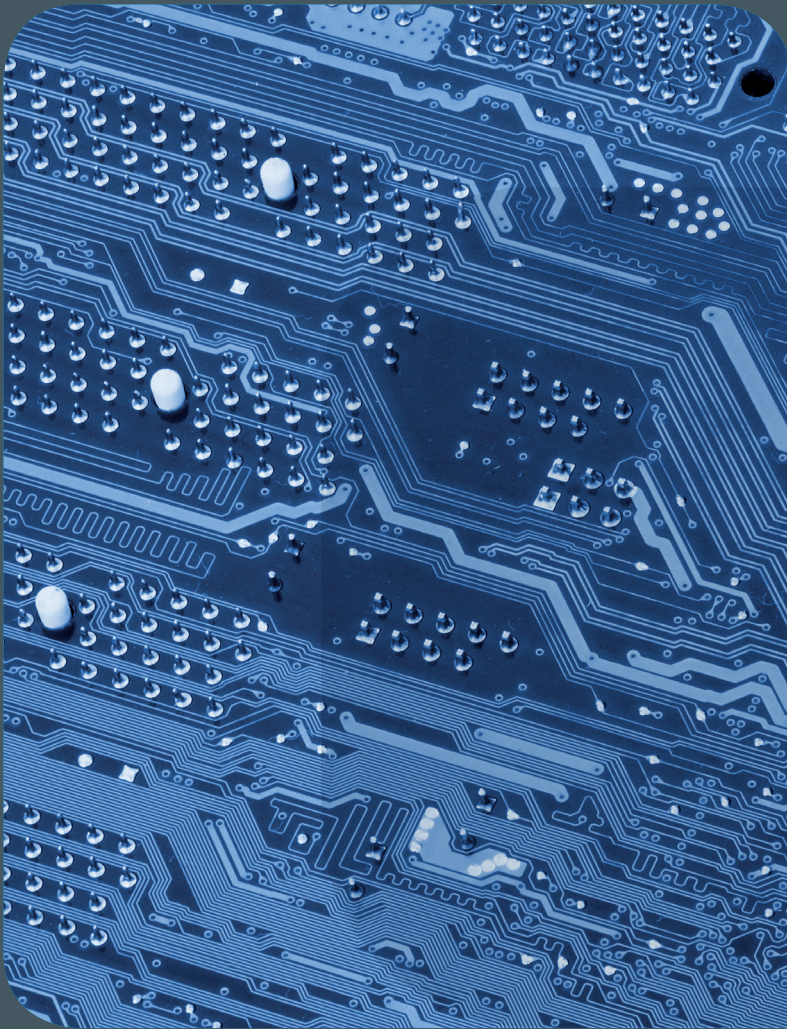


.NET



GWDG  
Gesellschaft für wissenschaftliche  
Datenverarbeitung mbH Göttingen





## GWDG NACHRICHTEN

# 09|22 Inhalt

.....

**4 Drucken und Scannen im neuen Göttinger Rechenzentrum** **7 Neue DNS-Infrastruktur bei der GWDG** **13 Task Dependent Workflow with Slurm** **17 Trends in Cloud Computing: KubeCon and OpenInfra 2022** **19 Kurz & knapp** **20 Personalia** **22 Academy**

### Impressum

.....

Zeitschrift für die Kunden der GWDG

ISSN 0940-4686  
45. Jahrgang  
Ausgabe 9/2022

**Erscheinungsweise:**  
10 Ausgaben pro Jahr

[www.gwdg.de/gwdg-nr](http://www.gwdg.de/gwdg-nr)

**Auflage:**  
550

**Fotos:**  
© Funtap - stock.adobe.com (1)  
© momius - Fotolia.com (6)  
© Fotogestoeber - Fotolia.com (12)  
© Robert Kneschke - Fotolia.com (22)  
© MPI-NAT-Medienservice (3)  
© GWDG (2, 5, 18, 20, 21)

**Herausgeber:**  
Gesellschaft für wissenschaftliche  
Datenverarbeitung mbH Göttingen  
Burckhardtweg 4  
37077 Göttingen  
Tel.: 0551 39-30001  
Fax: 0551 39-130-30001

**Redaktion:**  
Dr. Thomas Otto  
E-Mail: [thomas.otto@gwdg.de](mailto:thomas.otto@gwdg.de)

**Herstellung:**  
Ariane-Vivien Sternkopf  
E-Mail: [ariane-vivien.sternkopf@gwdg.de](mailto:ariane-vivien.sternkopf@gwdg.de)

**Druck:**  
Kreationszeit GmbH, Rosdorf



Prof. Dr. Ramin Yahyapour  
ramin.yahyapour@gwdg.de  
0551 39-30130

## *Liebe Kund\*innen und Freund\*innen der GWDG,*

*die angespannte Situation am Energiemarkt und die unklare Versorgungssituation im Herbst stellen alle vor große Herausforderungen. Auch bei der GWDG haben wir uns in den letzten Monaten mit Fragen zu den möglichen Auswirkungen beschäftigt und wie man diesen geeignet begegnet.*

*Dass IT signifikant Energie benötigt, dürfte kein Geheimnis sein. Energieeffizienz ist für uns kein neues Thema. Im neuen Rechenzentrum haben wir hierfür diverse Maßnahmen ergriffen. Auch wenn die GWDG am Göttingen Campus in die höchste Priorität eingestuft ist und somit die Versorgung gewährleistet sein sollte, betrachten wir natürlich, welche Dienste man in einer Notsituation ggf. reduzieren kann, um auch einen Beitrag zu leisten, die Versorgungssicherheit auch für andere zu gewährleisten.*

*Davon unbenommen sind die gestiegenen Energiekosten eine enorme Belastung, von denen wir noch gar nicht absehen können, wie dramatisch sich diese noch weiterentwickeln. Dies ist kein spezifisches Problem für die IT, trifft hier aber besonders zu. Ohne Unterstützung der Politik wird dies im Wissenschaftsbereich kaum zu bewältigen sein.*

**Ramin Yahyapour**

*GWDG – IT in der Wissenschaft*

# Drucken und Scannen im neuen Göttinger Rechenzentrum

**Text und Kontakt:**  
Dr. Konrad Heuer  
konrad.heuer@gwdg.de  
0551 39-30313

Nicht wenigen unserer Nutzer\*innen sind die Räumlichkeiten der GWDG am alten Standort im heutigen Max-Planck-Institut für Multidisziplinäre Naturwissenschaften, dem früheren Max-Planck-Institut für biophysikalische Chemie, noch vertraut. Unsere Großformatdrucker, der Großformatscanner und andere Peripheriegeräte fanden regen Zuspruch. Die Pandemie führte ab Frühjahr 2020 leider zur vorübergehenden Schließung des Rechenzentrums. In diese Zeit der Schließung fiel der Umzug der GWDG in das neue Göttinger Rechenzentrum. Nachfolgend sollen die entsprechenden neuen Räumlichkeiten mit den bekannten Peripheriegeräten vorgestellt werden.

## EINLEITUNG

Fast 50 Jahre lang war die GWDG in den Räumlichkeiten des früheren Max-Planck-Instituts für biophysikalische Chemie beheimatet, das zum 01.01.2022 mit dem Max-Planck-Institut für Experimentelle Medizin zum Max-Planck-Institut für Multidisziplinäre Naturwissenschaften fusioniert ist. Der dortige Rechnerraum hat viele Hardware-Generationswechsel erlebt, und auch die Räume für Kurse, Nutzer\*innen und Peripherie waren einem stetigen Wandel unterworfen. In den letzten Jahren zeigte sich jedoch immer mehr, dass die Räumlichkeiten für eine wachsende GWDG und eine wachsende Bedeutung von IT-Infrastruktur nicht mehr ausreichten. Ausweichstandorte für Personal und Hardware kamen hinzu, und schließlich einigten sich die Max-Planck-Gesellschaft und die Universität Göttingen mit zusätzlicher Beteiligung der Universitätsmedizin Göttingen auf einen Rechenzentrumsneubau, der nach längerer Standortsuche am Burckhardtweg 4 im nördlichen Randbereich der Universität und ein Stück unterhalb des Max-Planck-Campus entstand.

Aufgrund der verstärkten Zusammenarbeit der IT-Dienstleister auf dem Göttingen Campus sind über die GWDG hinaus auch Mitarbeiter\*innen der Abteilung G3-7 der Universitätsmedizin in das neue Gebäude eingezogen.

Das neue Rechenzentrum ist seit einiger Zeit geöffnet, und nachdem ein Großteil der notwendigen Vorarbeiten abgeschlossen worden ist, ist nun das Drucken und Scannen im neuen Gebäude möglich. Die Öffnungszeiten der GWDG montags bis freitags von 7:00 Uhr bis 21:00 Uhr und samstags und sonntags von 10:00 Uhr bis 18:00 Uhr haben sich nicht geändert. Wir empfehlen jedoch, zunächst bevorzugt den Zeitraum montags bis freitags von 9:00 Uhr bis 17:00 Uhr zu nutzen, in dem eine bessere Unterstützung seitens der GWDG möglich ist. Natürlich sind

die jeweils aktuell geltenden Pandemieregeln zu beachten. Die GWDG hält sich dabei selbstverständlich an die gesetzlichen Vorgaben, orientiert sich zusätzlich aber auch an den Regeln ihrer beiden Gesellschafter.

Die alten Räume der GWDG konnten trotz aller Wandlung nicht leugnen, dass sie zur Zeit des klassischen Großrechnerbetriebs geplant wurden und entstanden waren. Das neue Rechenzentrum bietet Besucher\*innen ein ganz anderes Bild.

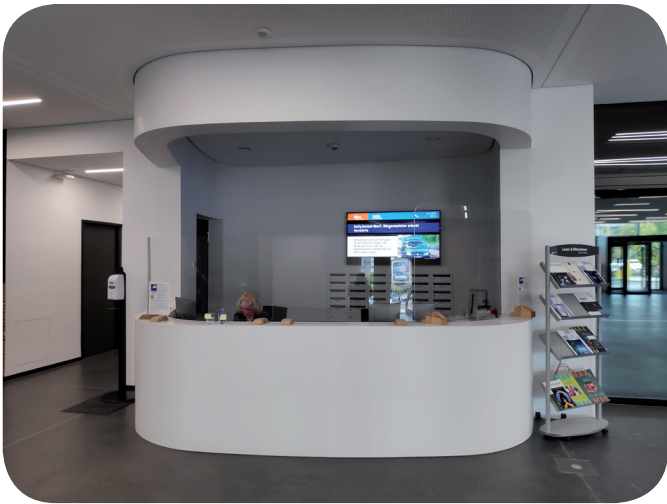
## DER ÖFFENTLICHE BEREICH

Im neuen Göttinger Rechenzentrum wird stärker als bisher zwischen öffentlichen und nicht frei zugänglichen Bereichen unterschieden, um höhere Anforderungen an den sicheren IT-Betrieb erfüllen zu können. Der Maschinenraumtrakt ist in besonderer Weise abgesichert, und auch die Büroräume der Mitarbeiter\*innen sind für Besucher\*innen nicht mehr ohne weiteres zugänglich.

### Printing and Scanning in the New Göttingen Data Center

More than a few of our users are still familiar with the premises of the GWDG at the old location in today's Max Planck Institute for Multidisciplinary Sciences, the former Max Planck Institute for biophysical Chemistry. Our large format printers, large format scanners and other peripheral devices were very well received. Unfortunately, the pandemic led to the temporary closure of the data center from spring 2020 on. During this period of closure, the GWDG moved to the new Göttingen data center. As far as the familiar peripheral devices are concerned, the new location will be presented in this article.





1\_Empfang im Eingangsbereich



3\_Farblaserdrucker im Druckerraum



2\_Der Beratungsraum

Der öffentliche Bereich erstreckt sich über Teile des Erd- und des Untergeschosses und umfasst im Kontext dieses Artikels

- den Eingangsbereich,
- den Druckerraum
- und einen Raum für IT-Spezialgeräte.

## DER EINGANGSBEREICH

Wer das Gebäude durch den Haupteingang betritt, findet sich in einem geräumigen Eingangsbereich wieder, an dessen Rückwand zurzeit als Erweiterung ein Hörsaal entsteht. Linkerhand ist der Empfang angesiedelt, der in seiner Funktion in etwa der alten Information der GWDG entspricht (siehe Abbildung 1). Die Mitarbeiter\*innen im Empfang stehen für Auskünfte gerne zur Verfügung. Sofern eine intensivere persönliche Beratung gewünscht ist, so kann diese nach Terminvereinbarung in einem speziellen Beratungsraum erfolgen (siehe Abbildung 2).

## DER DRUCKERRAUM

Der Druckerraum im Erdgeschoss ist wie am alten Standort mit zwei Großformatdruckern (siehe Abbildung 4) und einer passenden elektrischen Schneidemaschine ausgestattet. Hinzu kommen zwei leistungsfähige Farblaserdrucker, die im Lastverbund arbeiten und mit DIN-A3- und DIN-A4-Papier ausgerüstet sind



4\_Großformatdrucker im Druckerraum



5\_IT-Spezialgeräteaum

(siehe Abbildung 3). Die Druckgeschwindigkeit der Geräte reicht bis zu 70 Seiten pro Minute; angesteuert werden sie jeweils von einem EFI-Fiery-Controller.

Im Eingangsbereich werden an der Wand zum Druckerraum in den nächsten Wochen noch einige PC-Arbeitsplätze eingerichtet werden. Bis dahin empfehlen wir, zum Ausdrucken entweder den eigenen Arbeitsplatz zu verwenden und Ausdrucke später nur abzuholen oder ein Notebook mitzubringen, um im Rechenzentrum

arbeiten zu können.

Der Druckerraum ist während der Öffnungszeiten der GWDG in der Regel geöffnet. Falls in Einzelfällen der Raum verschlossen sein sollte, schließen die Mitarbeiter\*innen am Empfang den Druckerraum bei entsprechender Bitte gerne auf.

Weitere Informationen zur Nutzung der Drucker sind unter [https://docs.gwdg.de/doku.php?id=de:services:general\\_services:print\\_scan\\_services:print\\_server:start](https://docs.gwdg.de/doku.php?id=de:services:general_services:print_scan_services:print_server:start) zu finden.

## DER IT-SPEZIALGERÄTERAUM

Zwei Kursräume und ein Raum für IT-Spezialgeräte befinden sich Untergeschoss. Letzterer ist mit zwei PC-Arbeitsplätzen,

DIN-A3-Flachbettscanner, Diascanner und Großformatscanner sowie einigen Hilfsgeräten ausgerüstet (siehe Abbildung 5). Der IT-Spezialgeräte Raum eignet sich aufgrund seiner Ausstattung und Möblierung nicht nur für eine kurzzeitige Nutzung, sondern auch für umfangreichere Arbeiten.

Da der IT-Spezialgeräte Raum nur bei Bedarf aufgeschlossen wird, bitten wir alle an der Nutzung Interessierten, zunächst den Empfang im Erdgeschoss aufzusuchen. Empfehlenswert ist zudem eine Voranmeldung und Terminabsprache per E-Mail an [support@gwdg.de](mailto:support@gwdg.de), um Nutzungskonflikte zu vermeiden.

Das auf dem Foto im Außenbereich sichtbare Baumaterial ist dem noch nicht vollendeten zweiten Bauabschnitt des Bürotraktes geschuldet. ■



# FTP-Server

## Eine ergiebige Fundgrube!

### Ihre Anforderung

Sie möchten auf das weltweite OpenSource-Softwareangebot zentral und schnell zugreifen. Sie benötigen Handbücher oder Programmbeschreibungen oder Listings aus Computerzeitschriften. Sie wollen Updates Ihrer Linux- oder FreeBSD-Installation schnell durchführen.

### Unser Angebot

Die GWDG betreibt seit 1992 einen der weltweit bekanntesten FTP-Server mit leistungsfähigen Ressourcen und schneller Netzanbindung. Er ist dabei Hauptmirror für viele Open-Source-Projekte.

### Ihre Vorteile

- > Großer Datenbestand (65 TByte), weltweit verfügbar
- > Besonders gute Anbindung im GÖNET

- > Aktuelle Software inkl. Updates der gebräuchlichsten Linux-Distributionen
- > Unter pub befindet sich eine aktuell gehaltene locatedb für schnelles Durchsuchen des Bestandes.
- > Alle gängigen Protokolle (http, https, ftp und rsync) werden unterstützt.

### Interessiert?

Wenn Sie unseren FTP-Server nutzen möchten, werfen Sie bitte einen Blick auf die u. g. Webseite. Jede\*r Nutzer\*in kann den FTP-Dienst nutzen. Die Nutzer\*innen im GÖNET erreichen in der Regel durch die lokale Anbindung besseren Durchsatz als externe Nutzer\*innen.

>> [www.gwdg.de/ftp-server](http://www.gwdg.de/ftp-server)

# Neue DNS-Infrastruktur bei der GWDG

Text und Kontakt:  
Philipp Kreis  
philipp.kreis@gwdg.de

Wachsende Anforderungen an die DNS-Infrastruktur, sei es Verfügbarkeit, Leistungsfähigkeit oder Sicherheitsaspekte, haben die GWDG dazu bewogen, das Konzept ihrer DNS-Infrastruktur neu zu denken und nach heute gültigen Best Practices zu restrukturieren. Dieser Artikel soll einen kleinen Einblick in das neue DNS-Konzept der GWDG geben.

Das „Telefonbuch des Internets“. So oder so ähnlich wird der DNS-Dienst (Domain Name System) häufig beschrieben. Nicht verwunderlich, denn diese Analogie beschreibt den Verzeichnisdienst sehr passend. Es werden Namen zu Nummern übersetzt. Im Fall von DNS werden DNS-Namen (z. B. *www.gwdg.de*) zu IP-Adressen (134.76.9.48) übersetzt. Im Gegensatz zum klassischen Telefonbuch in Papierform, welches zunehmend an Bedeutung verliert, ist der DNS-Dienst aber zu einer unerlässlichen und tragenden Säule des Internets geworden. Die Nutzung von Computersystemen ist bei einem Ausfall des DNS praktisch nicht mehr möglich. Das DNS ist damit nicht nur obligatorisch, sondern auch systemrelevant und dominiert damit die Verfügbarkeit nahezu aller Dienste im IT-Bereich. Diese zentrale Bedeutung sowie die gestiegenen Anforderungen an das DNS haben den Ausschlag gegeben, das Konzept rund um den DNS-Dienst der GWDG zu überdenken und an moderne Strukturen anzupassen.

## DIE HISTORIE

Anfang der 90er Jahre wurden die ersten DNS-Server bei der GWDG in Betrieb genommen. Seinerzeit waren die lokalen Nameserver über die IP-Adressen 134.76.10.46 (*gwdu01.gwdg.de*) und 134.76.98.2 erreichbar. Im März 2003 wurde die IP-Adresse 134.76.33.21 (*ns2.gwdg.de*) auf einem neuen Linux-Server in der Fernmeldezentrale der Universität Göttingen in Betrieb genommen. Berichtet wurde darüber damals in einem Artikel in den GWDG-Nachrichten 2/2003 [1]. Der neue Server ersetzte die IP-Adresse 134.76.98.2 nach etwas mehr als einem Jahr Parallelbetrieb am 01.03.2004 vollständig. Auch hierzu gibt es einen GWDG-Nachrichten-Artikel [2].

Bis 2009 wurden aus einer eigenentwickelten Microsoft-Access-Datenbank die Konfigurationsdateien für die Nameserver erzeugt und auf die DNS-Server übertragen. Im Laufe der Zeit genügte dieses System aber nicht mehr den wachsenden Anforderungen. So wurde 2008, zusammen mit der Universität Göttingen, ein neues IP-Adressmanagement-System (IPAM) der Firma Blue-Cat Networks beschafft. In diesem Zuge wurde ein weiterer DNS-Server (*ns3.gwdg.de*) in Betrieb genommen. Abgesehen von den üblichen Hardware-Aktualisierungen wird dieses System bis heute eingesetzt.

## IST-ZUSTAND VOR DEM UMBAU

Das alte DNS-Konzept der GWDG (siehe Abbildung 1) beinhaltete neben dem Server für das webbasierte IP-Adressmanagement, also der zentralen IP- und DNS-Datenbank, drei DNS-Server: *gwdu01.gwdg.de* (134.76.10.46), *ns2.gwdg.de* (134.76.33.21) und *ns3.gwdg.de* (193.175.210.140). Letztgenannter wurde aus Gründen der Ausfallsicherheit außerhalb von Göttingen im GNZ [3] der MPG in Berlin aufgestellt.

Der Server *gwdu01.gwdg.de* (134.76.10.46) bestand aus zwei Servern im Failover-Betrieb. Die beiden übrigen Server waren jeweils Einzelserver ohne einen zusätzlichen Redundanz-Mechanismus. Die von den einzelnen Servern bereitgestellten Dienste sind in der Tabelle 1 dargestellt:

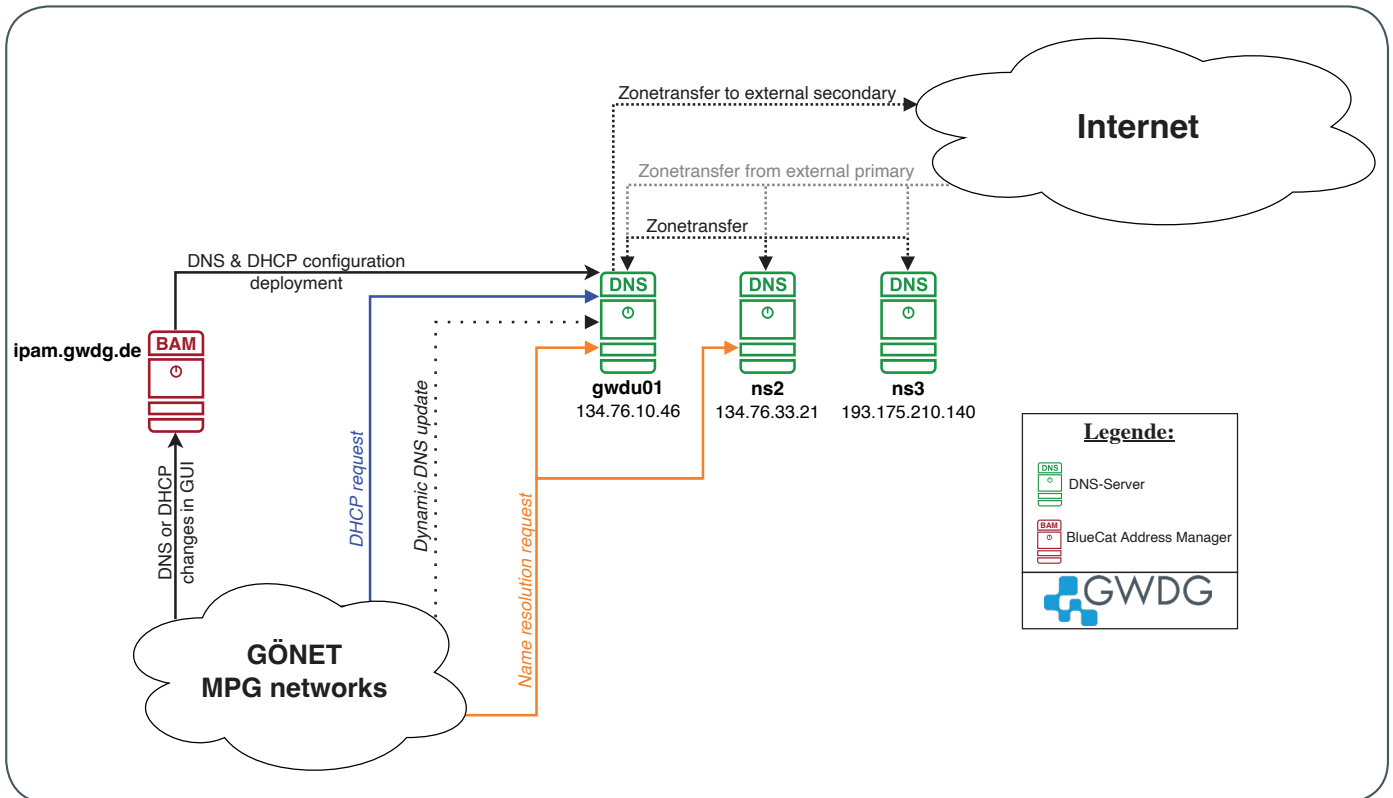
## MOTIVATION

Dieser dargestellte Aufbau hatte nach heutigem Stand der Technik und mit dem heutigen Wissen einige Nachteile und Sicherheitsrisiken, die wir mit der Restrukturierung lösen wollten.

## New DNS Infrastructure at the GWDG

Growing demands on the DNS infrastructure, be it availability, performance or security aspects, have prompted the GWDG to rethink the concept of the DNS infrastructure of the GWDG and to restructure it according to current valid best practices. The roles of authoritative nameserver, recursive nameserver and DHCP server were relocated to independent servers in a multi-stage restructuring phase, so that it is no longer possible for these roles to influence each other. Using a routing technique called „anycast“ all new resolvers are and will stay accessible via the well-known and established GÖNET nameserver IP addresses [4]. In addition, the metrics were improved and an automatism was implemented that selectively flushes zones from the cache of all resolvers when they have been changed on the authoritative nameservers. The transition went well and the past weeks and months have shown that the anycast solution scales well in everyday use.





1\_Alter Aufbau der DNS/DHCP-Landschaft

SERVER	DIENSTE
gwdu01.gwdg.de (134.76.10.46)	autoritativer Nameserver, rekursiver Nameserver (Resolver), DHCP-Server
ns2.gwdg.de (134.76.33.21)	autoritativer Nameserver, rekursiver Nameserver (Resolver)
ns3.gwdg.de (193.175.210.140)	autoritativer Nameserver

Tabelle 1: Dienste der DNS-Server vor dem Umbau

### Probleme beim Ausfall eines der beiden Resolver

In allen Betriebssystemen sollten immer zwei Nameserver („Resolver“ oder „Recurser“) eingetragen sein (siehe hierzu auch den Abschnitt „Grundeinstellungen“ auf der Informationsseite zur Endgerätekonfiguration [4]). Die Idee ist, dass bei einem Ausfall eines der beiden Nameserver der verbleibende gefragt werden kann. Die Vergangenheit hat uns leider gezeigt, dass dies in vielen Fällen nicht zufriedenstellend funktionierte. Manche Betriebssysteme haben sich nicht gemerkt, dass einer der Nameserver aktuell nicht erreichbar ist und so konnte es vorkommen, dass für jede neue DNS-Anfrage wieder zuerst der defekte Server gefragt wurde. Folglich musste jedes mal ein „Timeout“ abgewartet werden, um anschließend auf den noch funktionierenden Server auszuweichen. Für die Nutzer\*innen äußerte sich dieses Verhalten in einer gefühlt sehr langsamen Netzwerkverbindung und das, obwohl es mit dem eigentlichen Netzwerk keinerlei Probleme gab. Unsere Zielsetzung war folglich, dass beide IP-Adressen so weit wie möglich immer erreichbar sein sollten.

### Unterschiedliche Dienste und Rollen auf denselben Servern

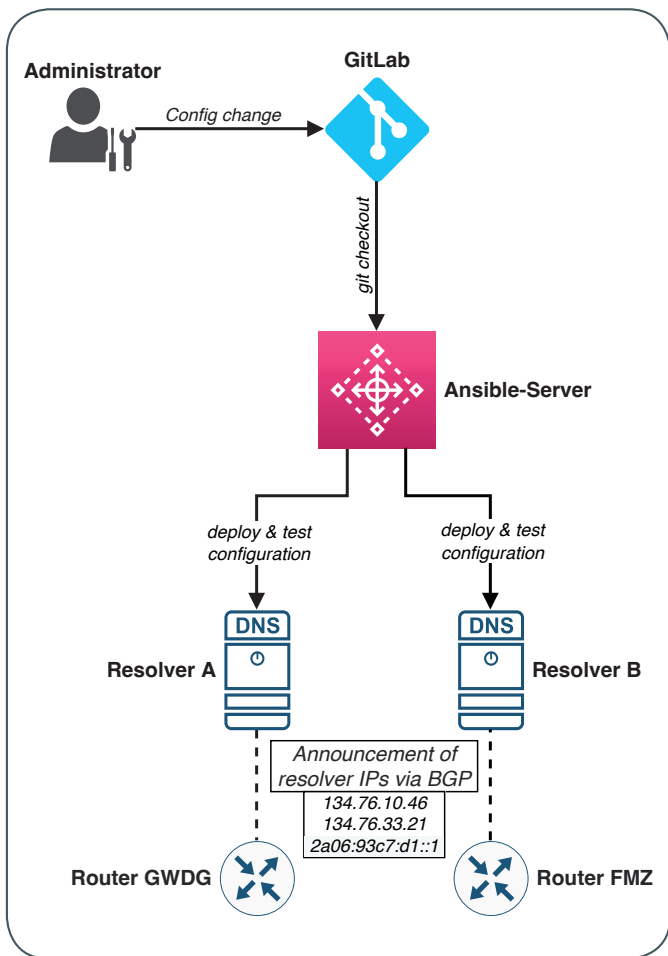
Die DNS-Server *gwdu01.gwdg.de* und *ns2.gwdg.de* haben beide autoritative sowie rekursive DNS-Anfragen beantwortet. Sie haben also jedem Anfragenden aus dem Internet gesagt, welche IP-Adresse *www.gwdg.de*, *www.uni-goettingen.de* oder

*www.mpg.de* hat, und auch für Anfragende aus dem GÖNET und den Max-Planck-Instituten versucht, z. B. die Adresse *vonejournal.jwd.example.com* oder jede andere Domain aufzulösen. Der DNS-Server *gwdu01.gwdg.de* war darüber hinaus der zentrale DHCP-Server [5] im GÖNET. Diese drei genannten Dienste bzw. Rollen haben eine unterschiedliche Funktionsweise und damit unterschiedliche Sicherheits-, Verfügbarkeits- und Geschwindigkeitsanforderungen. Probleme oder, im schlimmsten Fall, Angriffe auf einen der Dienste, könnten potenziell auch die anderen Dienste beeinflussen. Um die Dienste unabhängiger voneinander zu machen, sollten Sie aufgetrennt werden. [6]

### TECHNISCHE UMSETZUNG

Im ersten Schritt wurden von Ende 2019 bis etwa Mitte 2020 die rekursiven Nameserver sukzessive von den autoritativen DNS-Rollen befreit. Drei neue virtuelle BlueCat-DNS-Server (*dns1.gwdg.de*, *dns2.gwdg.de* und *dns3.gwdg.de*) beantworten seitdem der ganzen Welt die autoritativen DNS-Anfragen nach z. B. *meet.gwdg.de*. Die Konfiguration für diese drei neuen Server wird wie zuvor aus der IPAM-Datenbank generiert.

Im Anschluss daran erfolgte die Planung und Entwicklung eines neuen DNS-Resolvers. DNS-Resolver haben gegenüber autoritativen Nameservern und DHCP-Servern eine vergleichsweise einfache und statische Konfiguration. Ihre Aufgabe besteht schließlich nur darin, für den Client IP-Adressen herauszufinden und im besten Fall eine Zeit lang zwischenspeichern. Häufige Änderungen sind hier nicht zu erwarten. So bot es sich an, auf ein (kostspieliges) System von BlueCat zugunsten einer Eigenentwicklung zu verzichten. Als Grundlage wählten wir einen Mix aus einem Hardwareserver und einer virtuellen Maschine. Beide wurden mit Ubuntu Linux als Betriebssystem ausgestattet und bekamen eine minimale Grundkonfiguration. Jegliche weitere Konfiguration auf



2\_Funktionsprinzip der Resolverkonfiguration mithilfe von Ansible

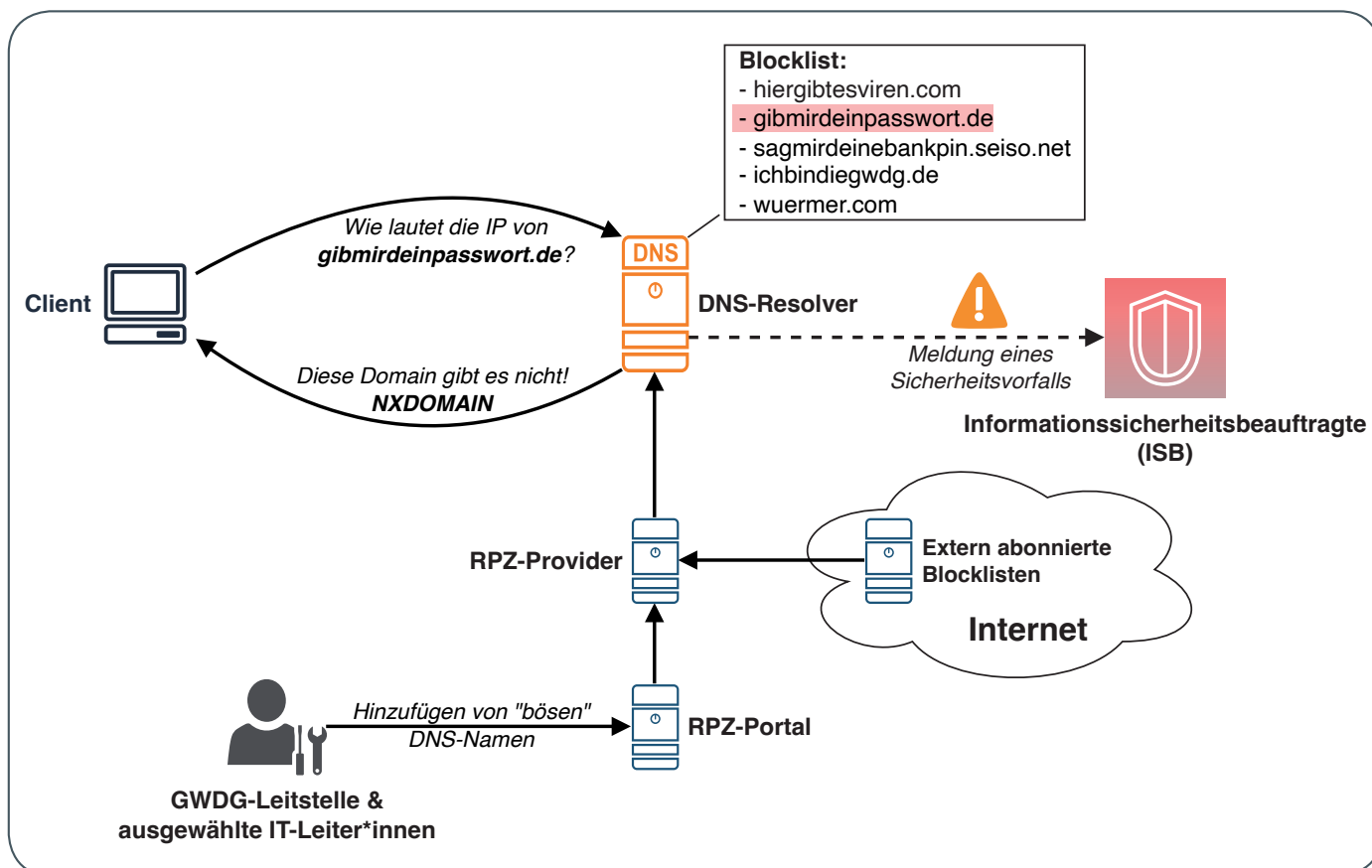
den Servern erfolgt mit dem Open-Source-Automatisierungswerkzeug Ansible.

Bei Ansible erfolgt die Konfiguration mithilfe sogenannter Playbooks. Diese Playbooks werden wiederum auf einer gewissen Anzahl von Servern, dem Inventar, ausgeführt. Dies hat den Vorteil, schnell und unkompliziert weitere Server, die genau gleich funktionieren, aufsetzen und in Betrieb nehmen zu können. Somit sind wir bei stark wachsenden Anfragen an unsere Resolver in der Lage, diese in der Anzahl zu skalieren. Die Konfigurationen für Ansible selber liegen auf einem speziell gesicherten GitLab und Notfallserver. Dieses ermöglicht ein einfaches und schnelles Disaster Recovery.

Das „Abspielden“ der Ansible Playbooks und die damit verbundene Konfiguration der Server erfolgt mithilfe eines zentralen Ansible-Servers. Dieser übernimmt alle Änderungen aus dem vorher erwähnten GitLab und bespielt damit die Server der Reihe nach (siehe Abbildung 2).

### Verbesserte Verfügbarkeit

Wie bereits erwähnt, ist es uns wichtig, Ausfälle einzelner Nameserver-IP-Adressen so gut es geht zu vermeiden. Um dies zu erreichen, setzen wir mit den neuen Resolvern auf die Adressierungsart „Anycast“. Bei Anycast erhalten mehrere gleichartige Server zusätzlich zu ihrer individuellen noch eine oder mehrere weitere IP-Adressen, die bei allen identisch sind. Die Router des Netzwerks sorgen dann dafür, dass eine Anfrage immer nur zu genau einem dieser Server geht. Dadurch sind nun alle unsere Resolver über alle bekannten Nameserver-IP-Adressen [4] erreichbar und nicht wie zuvor immer genau nur einer der Server. Ermöglicht wird dies durch das Routingprotokoll „BGP“, welches zwischen den Resolvern und dem jeweiligen Router „gesprochen“



3\_Funktionsprinzip der „DNS-Firewall“ (response policy zone)

wird. Sollte einer der Server ausfallen, so bekommen die verbleibenden Server entsprechend mehr Anfragen. Diese Konstruktion bietet also zeitgleich Ausfallsicherheit, eine Lastverteilung und eine ständige Prüfung der Funktionsfähigkeit, da im Normalfall alle Server zeitgleich in Betrieb sind. In klassischen Aktiv-Passiv-Lösungen passiert es hingegen schon mal, dass im Notfall die passive Einheit garnicht mehr richtig funktioniert oder die Übernahme der Funktion so lange dauert, dass es dennoch zu einer merklichen Verzögerung kommt.

„DNS-Firewall“

Bereits in der Vergangenheit waren die Nameserver Bestandteil des IT-Sicherheitskonzepts des GÖNET. So nutzen wir seit längerem die RPZ-Funktionalität (Response Policy Zone) dazu, Angriffe auf die Nutzer\*innen zu erschweren oder bestenfalls zu verhindern. Es soll konkret verhindert werden, dass „böse“ Webseiten überhaupt geöffnet werden können. Die Nameserver fungieren dazu als eine Art „DNS-Firewall“ (siehe Abbildung 3). Sie enthalten Blocklisten mit „bösen“ DNS-Namen, welche im Falle einer Abfrage durch einen Client mit einem „NXDOMAIN“ („Diesen Namen gibt es nicht“) beantwortet werden. Somit ist es den Angreifern nicht möglich, auf betrügerische Webseiten weiterzuleiten oder, bei einer Infektion, weitere Schadsoftware nachzuladen. Darüber hinaus erfolgt eine automatisierte Meldung an die zuständigen Informationssicherheitsbeauftragten. Diese können eventuellen Sicherheitsvorfällen direkt nachgehen, um so größere Schäden oder Infektionen abzuwenden.

Die GWDG abonniert dazu externe Blocklisten, die sehr zeitnah auf neue und sich verändernde Gefahren angepasst werden. Darüber hinaus werden GWDG-eigene Blocklisten durch die Leitstelle der GWDG und ausgewählte IT-Leiter\*innen gepflegt.

Zögern sie also nicht, verdächtige (Phishing-)E-Mails und Domains über unsere Support-Webseite unter <https://www.gwdg.de/support> oder per E-Mail an [support@gwdg.de](mailto:support@gwdg.de) zu melden.

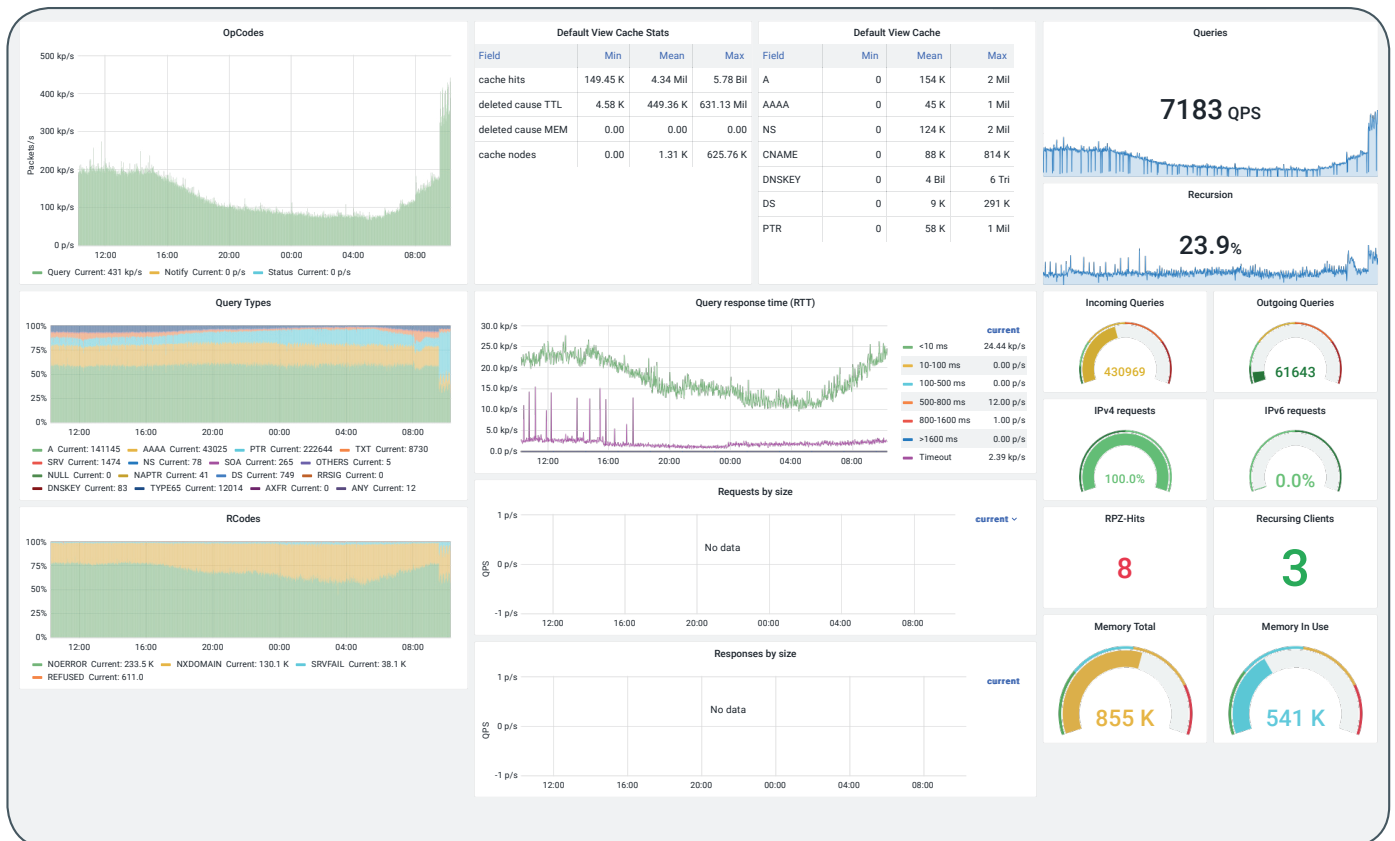
Erfassung von Metriken

Um einen reibungslosen Betrieb der Resolver gewährleisten zu können, wird ein umfassendes, aber dennoch einfach zu überblickendes Monitoring mit Metriken aus allen Bereichen benötigt. Ziel war es, Probleme oder Engpässe frühzeitig erkennen und beheben zu können. Neben den klassischen Parametern wie einem zentralen Syslog und dem Erfassen der Dienstverfügbarkeit werden zusätzlich alle relevanten DNS-Server-Metriken in eine InfluxDB geschrieben, die wiederum mit der webbasierten Anwendung Grafana visualisiert werden (siehe Abbildung 4).

DNS-Caching – Fluch und Segen zugleich

DNS-Caching ist ein probates Mittel, sehr viele rekursive DNS-Anfragen einfach handhaben zu können. So merkt sich der Resolver bei der ersten Anfrage für eine bestimmte Zeit, der TTL (Time To Live), die erfragte IP-Adresse zu dem DNS-Namen und kann diese bei einer erneuten Anfrage innerhalb dieser Zeit ohne Aufwand als Antwort sofort ausliefern.

Ein Nachteil, der sich mit dem Aufbau eigenständiger Resolver leider ergibt, ist, dass auch alle DNS-Namen, für die Änderungen direkt aus dem IPAM kommen, zwischengespeichert werden. So würde beispielsweise eine Änderung der IP-Adresse von [www.uni-goettingen.de](http://www.uni-goettingen.de) erst nach Ablauf der Speicherzeit (TTL) für alle Nutzer\*innen sichtbar sein. Um dies zu verhindern, wurde ein Automatismus implementiert, der geänderte DNS-Zonen aus den Resolver-Zwischenspeichern löscht. Somit wird der Resolver gezwungen, sofort und



4\_Übersichtsseite der Resolvermetriken



ungeachtet der verbleibenden TTL-Zeit erneut die Informationen über die DNS-Namen in unserem Beispiel in *uni-goettingen.de* zu erfragen. Dieser Vorgang erfolgt selektiv nur für die Zonen, in denen eine Änderung durchgeführt wurde.

Um dies zu realisieren, bedienen wir uns eines kleinen Tricks. Wir lassen uns von einem der autoritativen Nameserver ein DNS-Notify [7] zusätzlich an alle Resolver schicken. Die Resolver werten dieses Notify unter Zuhilfenahme von Skripten aus und löschen daraufhin alle lokalen Informationen (den „Cache“) zu dieser DNS-Zone.

## DER UMBAU

Nachdem die Entwicklung der neuen Resolver abgeschlossen war und das neue Konzept umfangreich, in Zusammenarbeit mit ausgewählten Instituten, getestet worden war, mussten sie abschließend in Betrieb genommen werden. Notwendig dafür war es, den DHCP-Dienst von der IP-Adresse 134.76.10.46 auf eine neue IP-Adresse umzuziehen. Um einen größeren Ausfall zu vermeiden, konnte dies nur in einer koordinierten, gleichzeitigen Aktion zusammen mit der Inbetriebnahme der Resolver erfolgen. Die DHCP-Server, weiterhin bestehend aus zwei Blue-Cat-Hardwareservern in einem Failover-Betrieb, wurden dazu neu konfiguriert und auf die neue IP gebracht. Damit war die IP-Adresse 134.76.10.46 „frei“ von anderen Aufgaben und konnte wenige Sekunden später auf die neuen Resolver übergehen. Die DNS/

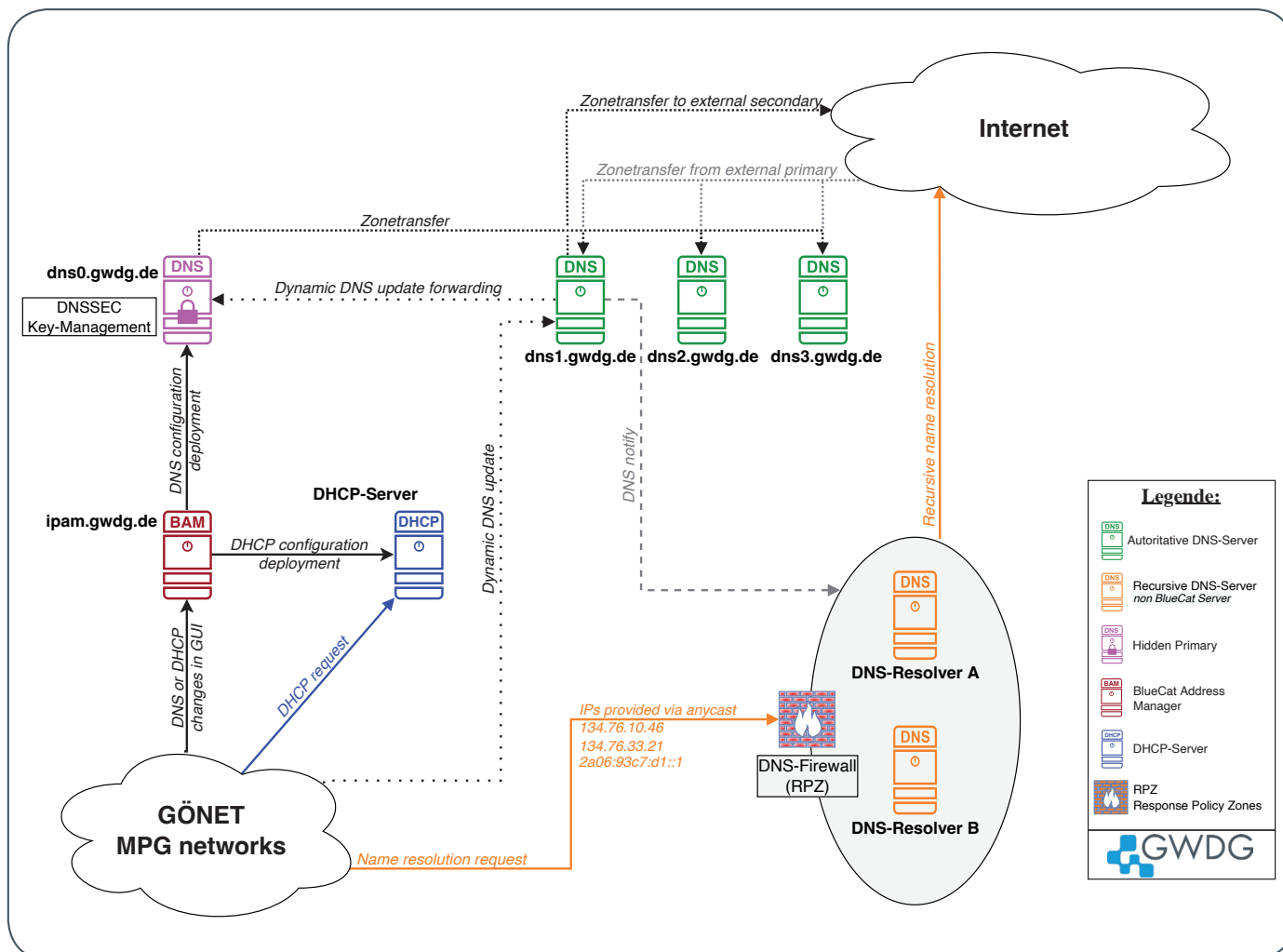
DHCP-Landschaft nach dem erfolgten Umbau ist in Abbildung 5 dargestellt, die Zuordnung der Dienste zu den DNS-Servern in Tabelle 2.

SERVER	DIENSTE
Resolver A & B (134.76.10.46 + 134.76.33.21)	rekursive Nameserver
DHCP-Server-Cluster	DHCP-Server
dns0.gwdg.de	Hidden Primary Nameserver
dns1.gwdg.de	autoritativer Nameserver
dns2.gwdg.de	autoritativer Nameserver
dns3.gwdg.de	autoritativer Nameserver in Berlin

Tabelle 2: Dienste der DNS-Server nach dem Umbau

## FAZIT

Mit der Einführung der neuen Resolver ist der größte Teil des Umbaus der DNS/DHCP-Landschaft der GWDG zunächst abgeschlossen. Wie bei Projekten in dieser Größe und in gewachsenen Umgebungen üblich, lief nicht immer alles nach Plan und problemlos. Insgesamt aber ist die Umstellung gut gelaufen und ging mit weniger Problemen einher als erwartet. In den vergangenen Wochen und Monaten konnten wir im Monitoring und anhand der guten Metriken erkennen, dass die Anycast-Lösung auch im Alltagsbetrieb hervorragend skaliert. Somit können wir uns nun den weiteren wichtigen Themen am Rande wie einem Resolver an den



Redundanz-Standorten, noch bessere DNS-Blocklisten, eine Überarbeitung der IPAM-Datenbank und weiteren Vereinfachungen im Workflow widmen.

Wir hoffen, wir konnten Ihnen einen Einblick in die Funktionsweise und die Neuerungen rund um den DNS-Dienst der GWGD bieten und Ihnen zeigen, dass der Vergleich mit einem Telefonbuch zwar durchaus passend erscheint, ein ausgewachsenes DNS-System aber ungleich wichtiger und komplexer ist.

## FUSSNOTEN

- [1] <https://www.gwdg.de/documents/20182/27257/gn0302.pdf>  
 [2] <https://www.gwdg.de/documents/20182/27257/gn0402.pdf>

- [3] Gemeinsames Netzwerkzentrum (GNZ) der Berlin-Brandenburgischen Max-Planck-Einrichtungen am Fritz-Haber-Institut der MPG e.V.  
 [4] [https://docs.gwdg.de/doku.php?id=de:services:network\\_services:goenet:konfiguration](https://docs.gwdg.de/doku.php?id=de:services:network_services:goenet:konfiguration)  
 [5] Der DHCP-Dienst weist Geräten beim Boot eine IP-Adresse zu.  
 [6] Hinzukommt ein technisches Detail im Bezug auf DNSSEC: Autoritative Server können Anfragen von Clients nicht validieren.  
 [7] DNS-Notifies werden gewöhnlich vom primären DNS-Server an alle seine sekundären DNS-Server versendet, um diesen mitzuteilen, dass es eine Änderung in der besagten Zone gibt, die diese sich dann abholen. ●



# Mailinglisten

## Mailversand leicht gemacht!

### Ihre Anforderung

Sie möchten per E-Mail zu oder mit einer Gruppe ausgewählter Empfänger\*innen kommunizieren, auch außerhalb Ihres Instituts. Sie möchten selbstständig eine Mailingliste verwalten, z. B. Empfänger\*innen hinzufügen oder entfernen. Bei Bedarf sollen sich auch einzelne Personen in diese Mailingliste einschreiben dürfen.

### Unser Angebot

Wir bieten Ihnen einen Listserver, der zuverlässig dafür sorgt, dass Ihre E-Mails an alle in die Mailingliste eingetragenen Mitglieder versendet werden. Die E-Mails werden automatisch archiviert. Das Archiv kann von allen Mitgliedern der Liste nach Schlagwörtern durchsucht werden. Die Anzahl Ihrer Mailinglisten ist unbegrenzt.

### Ihre Vorteile

- > Leistungsfähiges ausfallsicheres System zum Versenden von vielen E-Mails
- > Sie senden Ihre E-Mail lediglich an eine Mailinglisten-Adresse, die Verteilung an die Mitglieder der Mailingliste übernimmt der Listserver.

- > Listenmitglieder können an diese E-Mail-Adresse antworten. Eine Moderationsfunktionalität ist verfügbar, mit der Sie die Verteilung einer E-Mail genehmigen können.
- > Voller administrativer Zugriff auf die Einstellungen der Mailingliste und der Listenmitglieder
- > Obsolete E-Mail-Adressen werden vom System erkannt und automatisch entfernt.
- > Wenn Ihre E-Mail-Domäne bei uns gehostet wird, können Sie auch die Adresse der Mailingliste über diese Domäne einrichten lassen.

### Interessiert?

Für die Einrichtung einer Mailingliste gibt es zwei Möglichkeiten: Zum einen als registrierte\*r Nutzer\*in der GWGD im Selfservice über das Kundenportal der GWGD und zum anderen, indem Sie bitte eine entsprechende E-Mail an [support@gwdg.de](mailto:support@gwdg.de) senden, die die Wunsch-E-Mail-Adresse der Liste sowie die E-Mail-Adresse der Person, die die Liste bei Ihnen administrieren soll, enthalten sollte. Die administrativen Aufgaben sind leicht zu erlernen.

>> [www.gwdg.de/maillinglisten](http://www.gwdg.de/maillinglisten)

# Task Dependent Workflow with Slurm

**Text and Contact:**  
Pavan Kumar Siligam  
pavan.siligam@gwdg.de  
0551 39-30148

This article gives a primer on in-depth usage of slurm task dependency features done in a traditional way, highlighting some of the tedious workflow steps and complexities associated with this style of workflow and finally introducing “sworkflow” Python package which not only simplifies the workflow but also provides additional features enhancing user experience.

## INTRODUCTION

Scientific applications are usually embedded in a workflow which involves some pre-processing and post-processing steps in addition to executing the main application. Manually executing all stages of the workflow is tedious and tend to be more error prone. To overcome this, one approach is to automate the process by preparing a script or the other approach is to use popular workflow tools which are apt for the workflow needs. In practice, both approaches are widely used and such workflows setups are quite common in Earth Science or Bio Science communities. At the GWDG, the HPC team not only provides support for many workflow tools like snakemake, cylc, ecflow, etc used by GöHPC, NHR and DLR communities but also strives to simplify the process of creating custom workflows. In this article, the custom workflow setup using the slurm scheduler is presented.

Among all the features provided by a workflow tool, one prominent feature to immediately look at is defining “task dependency”. The tool looks at the user defined task dependency and constructs internally a Direct Acyclic Graph (DAG) which allows it to determine the task execution order and the scheduler which comes with the tool, executes the jobs in this order. Slurm, which is a workload manager for scheduling jobs on clusters, also attempts to solve the task dependent workflow by providing dependency directive. Users typically write a bash script or a Python script to utilize this feature. Regardless of the choice of scripting language used, the basic workflow style feels the same in any of the scripting languages.

## SLURM DEPENDENCY DIRECTIVE

Slurm provides *dependency* directive to define dependency between tasks. The syntax for defining the dependency is as follows:

```
--dependency=type:job_id[:job_id][:,type:job_id[:job_id]]
```

- *job\_id* is the *id* of the previously submitted job
- *type* is sort of a modifier to trigger this job based on previous job status (see table 1)

TYPE	DESCRIPTION
<i>after</i>	job starts or is cancelled
<i>afterok</i>	finished normally
<i>afternotok</i>	terminated non-zero
<i>afterany</i>	terminated in any way
<i>aftercorr</i>	after corresponding task in array
<i>singleton</i>	only one job sharing the same job name can run at a time

Table 1: Job dependency types supported by Slurm

## WORKFLOW

Based on the requirements of what needs to be achieved, a work-flow may have a simple **linear** (sequential) dependency among tasks or may **split** (multiple tasks depend on previous tasks) or **merge** (task depends on multiple previous tasks) or in any of these combinations.

### Linear

Consider the following simple example of linear dependency where task *C* depends on task *B* which in turn depends on task *A* (see figure 1).

The following code snippet (see code listing 1) shows the workflow of how this dependency between the tasks is defined.

## Aufgabenabhängiger Workflow mit Slurm

Wissenschaftliche Anwendungen sind in der Regel in einen Workflow eingebettet, der neben der Ausführung der Hauptanwendung einige Vor- und Nachverarbeitungsschritte umfasst. Die manuelle Ausführung aller Phasen des Workflows ist mühsam und tendenziell fehleranfälliger. Bei der GWDG bietet das HPC-Team nicht nur Unterstützung für viele Workflow-Tools wie Snakemake, cylc oder ecflow, die von GöHPC-, NHR- und DLR-Communities verwendet werden, sie ist auch bestrebt, den Prozess der Erstellung benutzerdefinierter Workflows zu vereinfachen. In diesem Artikel wird das benutzerdefinierte Workflow-Setup mit dem Slurm-Scheduler vorgestellt.



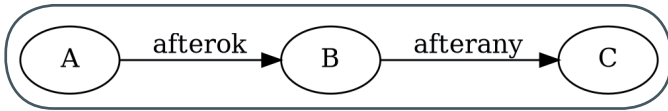


Figure 1: Linear dependency. Tasks A, B and C have linear dependency. The arrow direction indicates the execution order of the tasks

```
A_id=$(sbatch --parsable A.sh)
B_id=$(sbatch --parsable --depend=afterok:$A_id B.sh)
C_id=$(sbatch --parsable --depend=afterany:$B_id C.sh)
```

Code listing 1: Linear dependency

The process is started by submitting job A as it does not have any dependencies and it's job ID *A\_id* is captured. This job ID *A\_id* is used when submitting job B and B's job ID *B\_id* is in turn captured. And when submitting job C, *B\_id* is used. The *--parsable* flag formats the output just to show *jobid*.

The scripts *A.sh*, *B.sh* and *C.sh* may contain additional slurm directives or alternatively these directives may also be specified as command line arguments. Moreover these set of commands can be submitted in the shell interactively or from a bash script.

### Branching (Splitting)

Branching is one of the cases where two or more tasks depend on a previous task. In the following example, tasks C and D depend on task B which in turn depends on task A (see figure 2).

The coding part looks very similar to the linear dependency ones with a minor detail when submitting tasks C and D, task B's job ID *B\_id* is used as these tasks depend on it (see code listing 2).

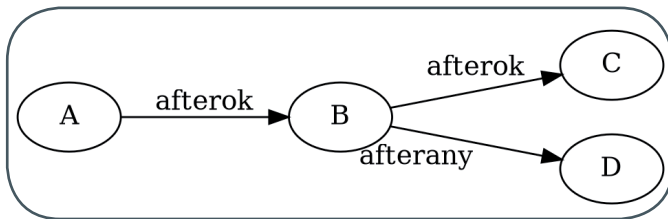


Figure 2: Branching. task C and task D depend on task B. The types *afterok* and *afterany* controls when these tasks are triggered.

```
A_id=$(sbatch --parsable A.sh)
B_id=$(sbatch --parsable --depend=afterok:$A_id B.sh)
C_id=$(sbatch --parsable --depend=afterok:$B_id C.sh)
D_id=$(sbatch --parsable --depend=afterany:$B_id D.sh)
```

Code listing 2: Branching. *B\_id* is used when submitting task C and D.

There is one subtle aspect when using these types *afterok*, *afternotok* and to understand it better, the following example can be considered. Similar to the linear dependency example, task C and D depend on B but only one of these tasks gets triggered based on the exit status of task B. So, if B successfully completes, then task C is triggered and if B fails then D is executed but not both. In this scenario, the task which is not triggered remains in the queue indefinitely unless explicitly cancelled (see figure 3).

The code is similar to the previous example with the minor change in the *types* declaration (see code listing 3.1).

The status of jobs in the queue can be checked using the *squeue* command which shows task D in *Pending* status and with the reason *DependencyNeverSatisfied* (see code listing 3.2).

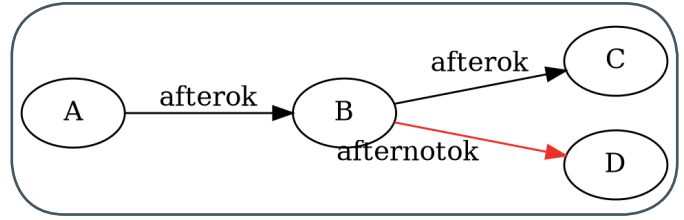


Figure 3: Branching. task C and task D depend on task B. The types *afterok* and *afternotok* control when these tasks are triggered. In this case either C or D gets executed depending on exit status of task B.

```
A_id=$(sbatch --parsable --wrap='sleep 2')
B_id=$(sbatch --parsable --depend=afterok:$A_id --wrap='sleep 2')
C_id=$(sbatch --parsable --depend=afterok:$B_id --wrap='sleep 2')
D_id=$(sbatch --parsable --depend=afternotok:$B_id --wrap='sleep 2')
echo "A: ${A_id}, B: ${B_id}, C: ${C_id}, D: ${D_id}"

A: 11646659, B: 11646660, C: 11646661, D: 11646662
```

Code listing 3.1: Branching. *B\_id* is used when submitting task C and D. All the jobs sleep for 2 seconds. The last line shows the mapping of task name to job id. In this case, task D is never triggered as the dependency type *afternotok* is never satisfied.

```
> squeue --me
JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)
11646662 medium wrap psiliga PD 0:00 1 (DependencyNeverSatisfied)
```

Code listing 3.2: Task D with job id (11646662) remains in the queue indefinitely as task B was successfully completed and *afternotok* is never satisfied.

### Merging (Joining)

Merging is one of the cases when a task depends on two or more previous tasks (see figure 4).

Again the coding part for such a dependency is quite flat and straight-forward. As task E depends on tasks C and D, these job IDs must be provided when submitting task E (see code listing 4).

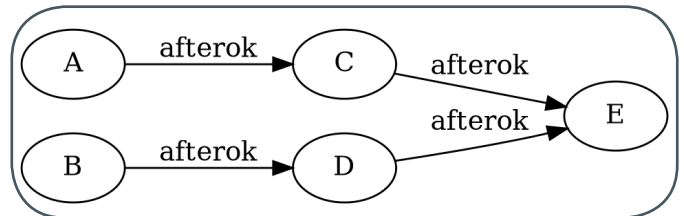


Figure 4: Merging. task E depends on task C and D.

```
A_id=$(sbatch --parsable A.sh)
B_id=$(sbatch --parsable B.sh)
C_id=$(sbatch --parsable --depend=afterok:$A_id C.sh)
D_id=$(sbatch --parsable --depend=afterok:$B_id D.sh)
E_id=$(sbatch --parsable --depend=afterok:$C_id:$D_id E.sh)
```

Code listing 4: Merging. As task E depends on C and D and so both ids are mentioned in dependency option.

Using figure 4 as reference, it is lot easier and straight-forward to write the coding part but in general one may start to think what the order of task submission should be. It turns out there are two possible ways which are defined as follows:

- Least dependent to most dependent tasks.
  - › One start with either A or B as they have no dependency
  - › then, either with C or D as their dependency is already submitted.
  - › Finally, E.

- Any task may be submitted whose dependency is already submitted.

Possible combinations in which the script can also be written (i.e., the ordering of lines):

```
A, B, C, D, E or B, A, C, D, E
A, B, D, C, E or B, A, D, C, E
A, C, B, D, E or B, D, A, C, E
```

The main point here is, there is a lot of flexibility in how the coding part is done in terms of the ordering of the tasks. For larger programs, writing the coding part will be a bit more extensive and thought intensive.

### Branching and Merging.

The workflow may involve both branching and merging of tasks (see figure 5 and code listing 5).

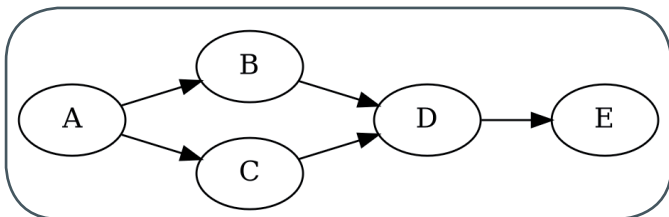


Figure 5: Branching and Merging. Tasks B and C branch off of A and they merge on task D.

```
A_id=$(sbatch --parsable A.sh)
B_id=$(sbatch --parsable --depend=afterok:$A_id B.sh)
C_id=$(sbatch --parsable --depend=afterok:$A_id C.sh)
D_id=$(sbatch --parsable --depend=afterok:$B_id:$C_id D.sh)
E_id=$(sbatch --parsable --depend=afterok:$D_id E.sh)
```

Code listing 5: Branching and merging.

### Discussion

So when one thinks of a work-ow, one has to consider the following aspects, namely: the order of submission, management of the job ID, that with large workflow the code can become messy, and that bash or Python scripting may eventually not be of much help with this style of workflow. To reduce the complexity, the workflow needs to be simplified some how. In the following section of this article, an alternative to this problem is presented.

Consider the example of branching and merging shown in figure 5, if the dependencies of all the tasks were to be written as mapping object in Python, it would look as follows (see code listing 6).

```
>>> dependency = {'B': 'A',
...               'C': 'A',
...               'D': ['B', 'C'],
...               'E': 'D'}
```

Code listing 6: Dependency mapping. It reads, B depends on A, C depends on A and so on....

This dependency mapping is a complete strip down version of the problem. Starting with this bare minimum dependency mapping, the next processing steps in order to prepare it for job submission are as follows:

- Resolve task ordering from dependency.
- Execute the tasks following this order.
- Fill in the *jobid*.

### Task ordering

To determine the ordering of the tasks, *TopologicalSorter* method from *graphlib* package is used, which comes as one of the packages in standard library in Python version 3.9 (see code listing 7).

```
>>> from graphlib import TopologicalSorter # python>=3.9
>>> def task_ordering(dependency):
...     return list(TopologicalSorter(dependency).static_order())
...
>>> dependency = {'A': 'B', 'C': 'A', 'D': ['B', 'C'], 'E': 'D'}
>>> ordering = task_ordering(dependency)
>>> print(ordering)
['A', 'B', 'C', 'D', 'E']
>>>
>>> # out-of-order construction of dependency mapping
>>> unordered = {k: dependency[k] for k in reversed(dependency)}
>>> unordered
{'E': 'D', 'D': ['B', 'C'], 'C': 'A', 'B': 'A'}
>>> task_ordering(unordered)
['A', 'C', 'B', 'D', 'E']
```

Code listing 7: *task\_ordering* function accepts dependency mapping to produce a sorted list of tasks. It can handle out-of-order defined dependency mapping.

The interesting thing about this one liner *task\_ordering* function is, it can handle dependency mapping defined in any order to produce a sort list of tasks that can be submitted. Which means the user does not need to worry about the task ordering any more and can stay more focused defining the dependencies between the tasks. Similar to this *task\_ordering* function, one can write small utility functions for handling *dependency types* (*after*, *afterok*, *afterany*..) and auto filling the job IDs etc. and put all these functions in small Python package. So, here is the experimental package called *sworflow* (see <https://github.com/siligam/sworflow>).

### SWORFLOW PACKAGE

Here is how the workflow looks like using *sworflow* using a Python interactive shell (see code listing 8 on the next site).

The first thing that can be noticed is both *dependency* and *jobs* are defined in separate dictionary mapping. Any task name missing the *jobs* mapping is assigned a default task of sleeping for 2 seconds. So in this case, *jobs* mapping can be an empty dictionary as well or can be completely omitted. The next thing is, the suite can be visualized. Submitting the suite returns the dictionary mapping of the task names with their job IDs. Further visualize include task name, job ID and the current status of the job.

### Command Line Interface

The *sworflow* package also comes with a command line interface. In this case, the dependency and jobs are defined in a *yaml* file (see code listing 9).

```
> cat jobarray.yaml
dependency:
  B: after:array
  C: afterok:array
  D: after:B+1
jobs:
  array: sbatch --array=5,10,15,20,25 --wrap='sleep $SLURM_ARRAY_TASK_ID'
```

Code listing 9: The suite definition file as a *yaml* file.

Few things to note in this file are as follows:

- The *array* job has 5 tasks.

```
>>> import sworkflow
>>>
>>> dependency = {'B': 'afterok:A',
...              'C': 'afterok:A',
...              'D': 'afterok:B:C',
...              'E': 'afterok:D'}
>>>
>>> jobs = {'A': "sbatch --wrap='sleep 2'",
...         'B': "sbatch --wrap='sleep 2'",
...         'C': "sbatch --wrap='sleep 2'",
...         'D': "sbatch --wrap='sleep 2'",
...         'E': "sbatch --wrap='sleep 2'"}
>>>
>>> s = sworkflow.Suite(dependency, jobs)
>>> s.visualize()
graph LR
    A[A] --> B[B]
    A --> C[C]
    B --> D[D]
    C --> D
    D --> E[E]
```

```
>>> s.submit()
{'A': '11959467', 'B': '11959468', 'C': '11959469', 'D': '11959470', 'E': '11959471'}
>>> s.visualize()
graph LR
    A["A 11959467 COMPLETED"] --> B["B 11959468 PENDING"]
    A --> C["C 11959469 PENDING"]
    B --> D["D 11959470 PENDING"]
    C --> D
    D --> E["E 11959471 PENDING"]
```

Code listing 8: Python session demonstrating sworkflow package usage

- Job B starts as soon as *array* job starts (*after*).
- Job C starts when *array* job successfully completes (*afterok*).
- Job D starts when job B starts with a delay of 1 minute (*after:B+1*).
- For missing job definitions (B, C, D), the default is *sbatch --wrap='sleep 2'*.

The following is the command line session of using sworkflow (see code listing 10).

The workflow using command line interface is quite similar to that of a Python session but a bit more flexible in the sense that an interactive session is not required. Additionally, it provides *status* command to check the status of the submitted suite and produces a nice table like output. For the *array* job status in the code listing. 10 shows 3C-2R which implies, 3 completed jobs and 2 running jobs.

## FINAL THOUGHTS

The sworkflow package is purely an experimental package created with the intention of exploring the slurm dependency features but in doing so, it opened up interesting possibilities like visualization of the suites, task name and task ID (job ID) association, keeping the focus on defining job dependency and not getting distracted by thinking about maintaining the order of task submission or managing the job IDs.

Looking at the bash scripting style of work-flow and sworkflow

```
> # visualization of the suite
> sworkflow -f jobarray.yaml vis
graph LR
    array[array] --> B[B]
    array --> C[C]
    B --> D[D]
```

```
> # submit
> sworkflow -f jobarray.yaml submit
{'array': '11959711', 'B': '11959712', 'C': '11959713', 'D': '11959714'}
```

```
> # status of the running suite
> sworkflow -f jobarray.yaml status
B 11959712 PENDING
C 11959713 PENDING
D 11959714 PENDING
array 11959711 5R
```

```
> # visualizing the current state of the suite
> sworkflow -f jobarray.yaml vis
graph LR
    array["array 11959711 1C-4R"] --> B["B 11959712 RUNNING"]
    array --> C["C 11959713 PENDING"]
    B --> D["D 11959714 PENDING"]
```

```
> # checking the current status of the suite
> sworkflow -f jobarray.yaml status
B 11959712 COMPLETED
C 11959713 PENDING
D 11959714 PENDING
array 11959711 3C-2R
```

Code listing 10: Command line work-flow of sworkflow package

package, it feels as if slurm provides the dependency features but the tooling part is left for the community to handle. ●

# Trends in Cloud Computing: KubeCon and OpenInfra 2022

**Text and Contact:**  
Samaneh Sadegh  
samaneh.sadegh@gwdg.de

In cloud computing, open-source software such as OpenStack and Kubernetes play a dominant role. Kubernetes streamlines operations, saves time and reduces costs for developing applications. On the other hand, OpenStack provides a scalable and flexible infrastructure for Kubernetes. A variety of open-source software has been built around OpenStack and Kubernetes which adds further value to this ecosystem. KubeCon and OpenInfra are two of the main conferences in cloud computing where open-source software developers and end users share their knowledge. In this article, we present a summary of the ones which were held on May and June 2022.

## INTRODUCTION

In recent years, the popularity of cloud and cloud native projects such as Kubernetes has increased significantly. Looking at some statistics, it can be seen that the computing model is changing:

- In 2020, Bultin announced that legacy architectures are transforming to containers and microservices [1].
- In 2021 SlashData reported that 5.6 million developers use Kubernetes, an increase of 67% over just one year [2].
- On April 2022, Gartner predicted that the worldwide public cloud revenue will grow by 20% in 2022 after a growth of 23% in 2021 [3].

Nowadays, when there is a need to create a highly available distributed application, Kubernetes (as the leading container-orchestration system) is the usual recommendation to use as a platform. Kubernetes runs efficiently on-premise as well as in a Cloud. The cloud version of Kubernetes often provides additional capabilities such as a better auto-scaling and fault tolerance. Therefore, the cloud environment is considered as the best infrastructure for a Kubernetes cluster.

Currently, GWDG provides a managed Kubernetes service on-premise (deployed on VMware vSphere) and a cloud server self-service (IaaS using OpenStack) with a plan of also providing Kubernetes hosted under OpenStack in the future. On this account, to be up-to-date with the newest developments in both ecosystems, we attended the two leading conferences in this area and are going to talk about some of the highlights and takeaways in this article:

- KubeCon + CloudNativeCon Europe 2022 Conference
- OpenInfra Summit 2022.

## KUBECON + CLOUDNATIVECON EUROPE 2022

“If you’re interested in enterprise IT infrastructure, Kubernetes should be your technology of choice, and KubeCon is the place to be.” (Jason Bloomberg, SiliconANGLE).

The Cloud Native Computing Foundation (CNCF) as part of the nonprofit technology consortium Linux Foundation provides support, oversight and direction for a plethora of fast-growing, cloud native projects. KubeCon + CloudNativeCon is the CNCF’s flagship conference where users, developers and newbies who are interested in cloud native technologies come together. While the first KubeCon in 2016 had around 700 people, KubeCon 2022 Europe had more than 7,000 people onsite and more than 10,000 joining virtually.

### Trends beim Cloud Computing: KubeCon und OpenInfra 2022

Beim Cloud Computing spielt Open-Source-Software wie OpenStack und Kubernetes eine wichtige Rolle. Kubernetes vereinfacht den Betrieb, spart Zeit und senkt die Kosten für die Entwicklung von Anwendungen. Auf der anderen Seite bietet OpenStack eine skalierbare und flexible Infrastruktur für Kubernetes. Um OpenStack und Kubernetes herum wurde eine Vielzahl von Open-Source-Software entwickelt, die diesem Ökosystem einen zusätzlichen Mehrwert verleiht. Und KubeCon und OpenInfra sind zwei der wichtigsten Konferenzen im Bereich Cloud Computing, auf denen Open-Source-Softwareentwickler und Nutzer\*innen ihr Wissen austauschen. In diesem Artikel präsentieren wir eine Zusammenfassung der wichtigsten Trends und Eindrücke von diesen Konferenzen, die im Mai und Juni 2022 stattfanden.





Figure 1: KubeCon and CloudNativeCon Europe 2022 – Valencia, Spain

During the five days (the main conference plus collocated events), there was a vast variety of talks, workshops, and presentations to attend. Here is an overview of what we took away from our experience at KubeCon + CloudNativeCon Europe 2022:

### Security

One of the main topics discussed in the conference was security and a secure software supply chain. Several large companies such as Red Hat, VMware, Cisco and others alongside with CNCF projects contributors and security solution providers like Pixie, Sysdig and Falco presented and discussed various aspects of this topic. Here are some hot takes from the security talks we attended:

- VMware Tanzu talked about the need of secure coding and how to use some frameworks such as SLSA and OWASP to insert security in a DevOps cycle.
- Apiiro talked about how security research is done giving a practical example of an attack using the “valueFiles” field in a helm chart of an app in Kubernetes.
- Red Hat, Pixie, Sysdig and Falco talked about detecting attacks and finding malicious containers by improving observability and applying AI/ML approaches.
- Cosmonic, Deepfence, Isovalent and Sysdig talked about how to enhance security using Kubernetes network policy, WebAssembly and eBPF. In addition, how to use Trivy to scan container images, Dockerfile and Kubernetes resource definition files.

### Observability

Observability was another popular topic with the following highlights of the talks we could attend:

- Trendyol talked about implementing a pipeline to improve observability for security in Kubernetes using Falco, Fluent-Bit, Prometheus and Grafana in order to detect threats at runtime using either normal scripts or AI/ML approaches.
- Red Hat and Timescale talked about Exemplars in Prometheus and how to use them in order to combine two of the main pillars of observability: metrics and tracing. This feature enables developers to do faster and more effective debugging in a distributed system.

- GitLab talked about service level objectives (SLOs) and how its metrics and rules can be defined using Prometheus and PromQL. The talks also explained a roadmap to move from metrics monitoring to observability.

### Further Topics

In addition, there were a variety of talks in different areas including but not limited to:

- The new features of different CNCF projects such as Kubernetes, Helm, Grafana, Harbor, etc.
- Data on Kubernetes (DOK) and how to better manage the storage.
- Service mesh and multi cluster failover.
- Enhancing the performance of both infrastructure and applications,

Last but not the least, the conference held several interesting workshops offering hands-on experience in Pod security, Kubernetes Networking, Debugging with kubectl, and GitOps. Almost all the sessions were recorded and are available in CNCF channel [4].

### OPENINFRA SUMMIT 2022

On October 2020, OpenStack foundation transformed into OpenInfra foundation to cover a wider variety of open-source technologies. Cloud native projects require a programmable infrastructure, and OpenInfra’s objective is to provide open source solutions for that. Currently more than 180 data centers which provide public cloud services are based on OpenStack which is almost 27% of the market share.

OpenInfra summit is the foundation’s annual event where open-source projects such as OpenStack, Kata Containers, Zuul, Magma and others present their solutions in various categories including cloud, HPC, hardware enablement, AI/ML, CI/CD, security, etc.

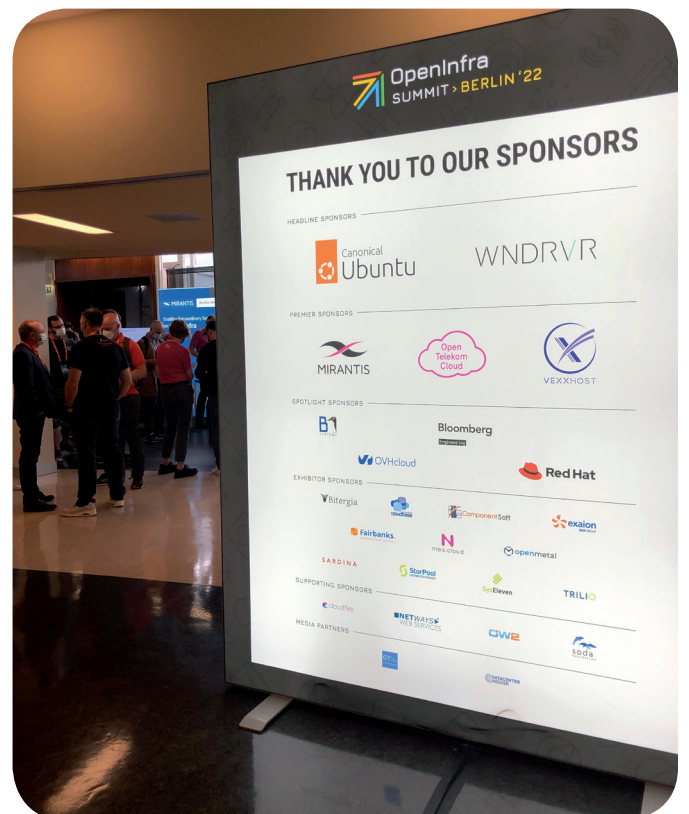


Figure 2: OpenInfra Summit 2022 – Berlin, Germany

Here is an overview of what we took away from OpenInfra summit 2022:

### Container Infrastructure and OpenStack Cloud

Container infrastructure and OpenStack cloud were two of the main topics discussed in the summit. Here are some key takeaways:

- Red Hat, OVHCloud, and Ericsson talked about the configuration and improvements of different components in OpenStack such as Neutron and Nova.
- CanaryBit, SecuStack, and Rivos talked about confidential computing and Kata containers as a security solution to protect workloads (code and data) from infrastructure providers.
- Canonical, Red Hat, and OSISM talked about different experiences they had for integrating OpenStack with Kubernetes and their solutions for auto-scaling, backup, and security enhancement.
- Open Telekom Cloud talked about how they have utilized OpenStack and other open-source solutions to provide a public cloud in Europe where the Corona warning application is running.

### GitOps and CI/CD

GitOps and CI/CD was another topic discussed in the talks we could attend with the following highlights:

- Red Hat talked about CI/CD concepts for the ones who are new to this topic. They ran through a series of common tasks and steps which are handled by CI/CD pipelines.
- Open Telekom Cloud talked about different open-source tools they have used to build their environment including Kubernetes, Terraform, Helm and ArgoCD.

- Volvo, BMW, Workday, Acme Gating, and the main developers presented their experience with Zuul in different talks. Zuul is an open-source project gating system (like a CI/CD system but more so) which originally was developed for OpenStack and now is used by many projects.

### Further Topics

In addition, there were a variety of talks in different areas including but not limited to:

- Application of AI/ML methods in infrastructure area such as anomaly detection, improving CI/CD in Kubernetes ecosystem, etc.
- How OpenStack can be deployed and configured to better fit HPC and big memory workloads.
- Edge deployment of Kubernetes and OpenStack.
- Cloud storage and Ceph.

Almost all the sessions were recorded and are available in Open Infrastructure Foundation channel [5].

### LINKS

- [1] <https://builtin.com/software-engineering-perspectives/enterprise-cloud-native-kubernetes>
- [2] <https://www.cncf.io/blog/2021/12/20/new-slashdata-report-5-6-million-developers-use-kubernetes-an-increase-of-67-over-one-year/>
- [3] <https://www.gartner.com/en/newsroom/press-releases/2022-04-19-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-500-billion-in-2022#:~:text=IaaS%2C%20DaaS%20and%20PaaS%20to,latest%20forecast%20from%20Gartner%2C%20Inc>
- [4] <https://www.youtube.com/c/cloudnativefdn/playlists>
- [5] <https://www.youtube.com/c/OpenStackFoundation/playlists>

## Kurz & knapp

### Erreichbarkeit der GWDG am Tag der Deutschen Einheit

Die Service-Hotline der GWDG ist am 03.10.2022, dem Tag der Deutschen Einheit, telefonisch nicht erreichbar.

Falls Sie sich an diesem Tag an die GWDG wenden möchten, erstellen Sie bitte eine Anfrage über unsere Support-Webseite

unter <https://www.gwdg.de/support> oder schicken eine E-Mail an [support@gwdg.de](mailto:support@gwdg.de). Das dahinter befindliche Ticket-System wird auch an diesem Tag von Mitarbeiter\*innen der GWDG regelmäßig überprüft. Wir bitten alle Nutzer\*innen, sich darauf einzustellen.

Pohl

## NEUER MITARBEITER TREVOR KHWAM TABOUGUA

Seit dem 1. August 2022 ist Herr Trevor Khwam Tabougua als wissenschaftlicher Mitarbeiter in der Arbeitsgruppe „Computing“ (AG C) tätig. Er unterstützt dort das Datenmanagement-Team und die Installation von Spack auf dem HLRN-IV-System „Emmy“. Herr Khwam Tabougua begann sein Studium in Tunesien an der Universität Gabès, wo er seinen Bachelor-Abschluss in Mathematik erwarb. Anschließend setzte er sein Mathematik-Studium mit Schwerpunkt Stochastik an der Universität Göttingen fort, das er erfolgreich mit dem Master abschloss. Während seines Studiums arbeitete er als studentische Hilfskraft am Fraunhofer Institut für Energiewirtschaft und Energiesystemtechnik (IEE) in Kassel und am Institut für Mathematische Stochastik (IMS) der Universität Göttingen, wo er Erfahrungen in verschiedenen Bereichen wie Data Science, maschinelles Lernen und Python-Programmierung sammeln konnte. Herr Khwam Tabougua ist per E-Mail unter [trevor-khwam.tabougua@gwdg.de](mailto:trevor-khwam.tabougua@gwdg.de) zu erreichen.



Kunkel

## NEUE MITARBEITER\*INNEN ALEXANDRA PHILINE MERTEN, MAXIMILIAN WILHELM SCHEID UND HENRI WAGNER

Seit dem 20. Juli 2022 verstärken Frau Alexandra Philine Merten, Herr Maximilian Wilhelm Scheid und Herr Henri Wagner das Support-Team im Helpdesk der GWDC als wissenschaftliche bzw. studentische Hilfskraft. Ihre Haupttätigkeiten sind der First-Level-Support, wo sie Anfragen ratsuchender Nutzer\*innen entgegennehmen, diesen direkt helfen oder die Anfragen an Kolleg\*innen im Second-Level-Support zur weiteren Bearbeitung übergeben. Alle neuen Hilfskräfte studieren zurzeit an der Georg-August-Universität Göttingen: Frau Merten im Fach Wirtschaftsinformatik, Herr Scheid im Fach Politikwissenschaft und Herr Wagner im Fach Physik.

Helmvoigt





## NEUER MITARBEITER RASMUS GÖRNANDT

Seit dem 1. August 2022 ist Herr Rasmus Görnandt in der Arbeitsgruppe „Nutzer-service und Betriebsdienste“ (AG H) tätig. Er unterstützt im Bereich der Leitstelle des Göttinger Rechenzentrums. Herr Görnandt absolvierte zunächst eine Ausbildung zum Bürokaufmann. Seinen Diplom-Abschluss als Wirtschaftsjurist erlangte er an der Fachhochschule Braunschweig/Wolfenbüttel. Seine Spezialgebiete waren Personalwirtschaft und Arbeitsrecht. Herr Görnandt bringt seine zuletzt langjährige Erfahrung im Sicherheitsdienst jetzt in die Leitstelle im Göttinger Rechenzentrum ein. Er ist per E-Mail unter [rasmus.goernandt@gwdg.de](mailto:rasmus.goernandt@gwdg.de) zu erreichen.



Pohl



## NEUE AUSZUBILDENDE MARIA ELISABETH HANE BUTT UND VLADISLAV MUTRUK

Am 1. August 2022 hat Frau Maria Elisabeth Hanebutt ihre Ausbildung zur Fachinformatikerin (IHK) in der Fachrichtung Anwendungsentwicklung bei der GWDG begonnen. Frau Hanebutt hat das Berufliche Gymnasium Eschwege besucht und dort den Schwerpunkt Technische Informatik gewählt. Nach Vollendung der gymnasialen Oberstufe hat sie die Voraussetzungen erfüllt, um mit Abschluss ihrer Ausbildung zur Fachinformatikerin die Fachhochschulreife zu erlangen. Frau Hanebutt ist per E-Mail unter [maria.hanebutt@gwdg.de](mailto:maria.hanebutt@gwdg.de) zu erreichen.

Lewandrowski



Am 1. August 2022 hat Herr Vladislav Mutruk seine Ausbildung zum Fachinformatiker (IHK) in der Fachrichtung Systemintegration bei der GWDG begonnen. Herr Mutruk hat nach seinem Abitur in Weimar ein Studium des Wirtschaftsingenieurwesens an der Leibniz-Universität Hannover begonnen und ist 2019 an die Georg-August-Universität Göttingen in den Studiengang Wirtschaftsinformatik gewechselt. Dort sammelte er erste Kenntnisse im Bereich der Informatik und entschied sich wegen des stärkeren Praxisbezugs für eine Ausbildung zum Fachinformatiker bei der GWDG. Herr Mutruk ist per E-Mail unter [vladislav.mutruk@gwdg.de](mailto:vladislav.mutruk@gwdg.de) zu erreichen.

Kopp





**INFORMATIONEN:**  
support@gwdg.de  
0551 201-1523

September bis  
Dezember 2022

# Academy

KURS	DOZENT*IN	TERMIN	ANMELDEN BIS	AE
WORKING WITH GRO.DATA	Király	13.09.2022 10:00 – 11:30 Uhr	12.09.2022	0
ARBEITEN MIT GRO.PLAN	Gnadt	15.09.2022 14:00 – 15:30 Uhr	14.09.2022	0
AFFINITY DESIGNER – GRUNDKURS	Töpfer	27.09.2022 10:00 – 15:00 Uhr	20.09.2022	4
INTRODUCTION TO PERFORMANCE ENGINEERING	Dr. Ogaja	29.09.2022 9:30 – 16:00 Uhr	22.09.2022	4
WORKING WITH GRO.DATA	Király	11.10.2022 10:00 – 11:30 Uhr	10.10.2022	0
ADMINISTRATION VON WINDOWS-RECHNERN IM ACTIVE DIRECTORY DER GWDC	Quentin	12.10.2022 9:00 – 12:30 und 13:30 – 15:30 Uhr	05.10.2022	4
HYBRID IDENTITY – INTEGRATION DES ACTIVE DIRECTORY IN MICROSOFT AZURE ACTIVE DIRECTORY	Kopp	25.10. – 26.10.2022 9:00 – 12:00 und 13:00 – 16:00 Uhr	18.10.2022	8
ARBEITEN MIT GRO.PLAN	Gnadt	01.11.2022 10:00 – 11:30 Uhr	31.10.2022	0
GRUNDLAGEN DER BILDBEARBEITUNG MIT AFFINITY PHOTO	Töpfer	02.11. – 03.11.2022 9:30 – 16:00 Uhr	26.10.2022	8
USING THE GWDC SCIENTIFIC COMPUTE CLUSTER – AN INTRODUCTION	Kellner	07.11.2022 9:30 – 16:00 Uhr	31.10.2022	4

KURS	DOZENT*IN	TERMIN	ANMELDEN BIS	AE
PARALLEL PROGRAMMING WITH MPI	Prof. Haan	08.11. – 09.11.2022 9:15 – 17:00 Uhr	01.11.2022	8
WORKING WITH GRO.DATA	Király	08.11.2022 10:00 – 11:30 Uhr	07.11.2022	0
PROGRAMMING WITH CUDA – AN INTRODUCTION	Prof. Haan	15.11.2022 9:15 – 17:00 Uhr	08.11.2022	4
INDESIGN – GRUNDLAGEN	Töpfer	15.11. – 16.11.2022 9:30 – 16:00 Uhr	08.11.2022	8
EINFÜHRUNG IN DIE STATISTISCHE DATENANALYSE MIT SPSS	Cordes	23.11. – 24.11.2022 9:00 – 12:00 und 13:00 – 15:30 Uhr	16.11.2022	8
ANGEWANDTE STATISTIK MIT SPSS FÜR NUTZER*INNEN MIT VORKENNTNISSEN	Cordes	07.12. – 08.12.2022 9:00 – 12:00 und 13:00 – 15:30 Uhr	30.11.2022	8
WORKING WITH GRO.DATA	Király	13.12.2022 10:00 – 11:30 Uhr	12.12.2022	0
ARBEITEN MIT GRO.PLAN	Gnadt	15.12.2022 14:00 – 15:30 Uhr	14.12.2022	0

#### Teilnehmerkreis

Das Angebot der GWDG Academy richtet sich an die Beschäftigten aller Einrichtungen der Universität Göttingen, der Max-Planck-Gesellschaft sowie aus wissenschaftlichen Einrichtungen, die zum erweiterten Kreis der Nutzer\*innen der GWDG gehören. Studierende am Göttingen Campus zählen ebenfalls hierzu. Für manche Kurse werden spezielle Kenntnisse vorausgesetzt, die in den jeweiligen Kursbeschreibungen genannt werden.

#### Anmeldung

Für die Anmeldung zu einem Kurs müssen Sie sich zunächst mit Ihrem Benutzernamen und Passwort im Kundenportal der GWDG (<https://www.gwdg.de>) einloggen. Wenn Sie zum Kreis der berechtigten Nutzer\*innen der GWDG gehören und noch keinen GWDG-Account besitzen, können Sie sich im Kundenportal unter dem URL <https://www.gwdg.de/registration> registrieren. Bei Online-Kursen kann das Anmeldeverfahren abweichen. Genauere Informationen dazu finden Sie in der jeweiligen Kursbeschreibung. Einige Online-Angebote stehen Ihnen jederzeit und ohne Anmeldung zur Verfügung.

#### Absage

Absagen können bis zu sieben Tagen vor Kursbeginn erfolgen. Bei kurzfristigeren Absagen werden allerdings die für den Kurs angesetzten Arbeitseinheiten (AE) vom AE-Kontingent der jeweiligen Einrichtung abgezogen.

#### Kursorte

Aufgrund der aktuellen Corona-Situation finden zurzeit nahezu alle Kurse in einem geeigneten Online-Format und nicht als Präsenzkurse statt. Nähere Informationen dazu finden Sie bei den jeweiligen Kursen. Auf Wunsch und bei ausreichendem Interesse führen wir auch Kurse vor Ort in einem Institut durch, sofern dort ein geeigneter Raum mit entsprechender Ausstattung zur Verfügung gestellt wird.

#### Kosten bzw. Gebühren

Die Academy-Kurse sind – wie die meisten anderen Leistungen der GWDG – in das interne Kosten- und Leistungsrechnungssystem der GWDG einbezogen. Die den Kursen zugrundeliegenden AE werden vom AE-Kontingent der jeweiligen Einrichtung abgezogen. Für alle Einrichtungen der Universität Göttingen und der Max-Planck-Gesellschaft sowie die meisten der wissenschaftlichen Einrichtungen, die zum erweiterten Kreis der Nutzer\*innen der GWDG gehören, erfolgt keine Abrechnung in EUR. Dies gilt auch für die Studierenden am Göttingen Campus.

#### Kontakt und Information

Wenn Sie Fragen zum aktuellen Academy-Kursangebot, zur Kursplanung oder Wünsche nach weiteren Kursthemen haben, schicken Sie bitte eine E-Mail an [support@gwdg.de](mailto:support@gwdg.de). Falls bei einer ausreichend großen Gruppe Interesse besteht, könnten u. U. auch Kurse angeboten werden, die nicht im aktuellen Kursprogramm enthalten sind.



Gesellschaft für wissenschaftliche  
Datenverarbeitung mbH Göttingen