

OpenSource

Volume: 11 | Issue: 03 | Pages: 100 | January 2023

THE COMPLETE MAGAZINE ON OPEN SOURCE

ForYou

An **EFY** GROUP Publication

Containers

How To Use Them Well

Docker SBOM: Keeping
A Check On Security Risks

Deploying A Java App
In Kubernetes

The Benefits Of
Containerization

Setting Up A
Docker Container



AI: An Introduction To Scikit-learn
And Our First Trained Model



Podman: The Pod
Manager Tool

Wanna Be Your Own Boss?

DO OPEN SOURCE. ←



Demand for Open Source is sky rocketing. Be it for managing IT infrastructure or development of software--Open Source solutions are what customers are seeking.

All you need to do is develop expertise in an Open Source stack, and then build a team around it!

And, Open Source For You can be your friend and a guide through this journey.

TO READ OUR PRINT EDITION Visit: <https://subscribe.efyindia.com>

TO READ OUR EZINE EDITION Visit: <https://ezine.lfymag.com>

WORLD'S LEADING PUBLICATION ON OPEN SOURCE

Looking for marketing solutions to engage with cutting edge techies?
Contact us at growmybiz@efy.in OR call us at +91-9811155335.



Does Your Antivirus Solution Provide You With Complete Protection?

Don't RELY on 20 years old technology to FIGHT current date viruses



TOP 10 things that your Antivirus solution should provide to tackle today's threat...

Ransomware File Protection

ATP- Advance Threat Protection

EDR- Endpoint
Detection And Response

Application Blocking

Deep Learning
Malware Analysis

Exploit Prevention

URL Blocking

Disk and Boot
Record Protection

Peripheral Control

Respond Investigate Remove
(Root Cause Analysis)

If it doesn't, contact us for a solution which can

Contact: Santosh on 9971696319 or Email at santosh.gupta@itsipl.com

ITS
A Total Solutions Company

I. T. Solutions India Private Limited

D-88/5, Okhla Industrial Area, Okhla Phase I, New Delhi -110020
Ph: 011-47695000 • Email: sales@itsipl.com • www.itsipl.com

• Mumbai
• Jaipur
• Chandigarh

SOPHOS 2020 BEST NEXTGEN PARTNER | Delhi/ NCR
SOPHOS 2019 BEST NEXTGEN PARTNER | Delhi/ NCR
SOPHOS 2018 BEST SI PARTNER | Delhi

FOR U & ME

23 How to Create PDFs and DOCX Files for Free

30 AI Based Tools that Aid Research

FOCUS

44 Deploying a Java App in Docker

57 Getting Started with Kubernetes: The Ultimate Guide

65 Docker SBOM: Keeping a Check on Security Risks

70 The Benefits of Containerization

73 Podman: The Pod Manager Tool

77 Setting Up a Docker Container

80 Why Developers Opt for Containers

DEVELOPERS

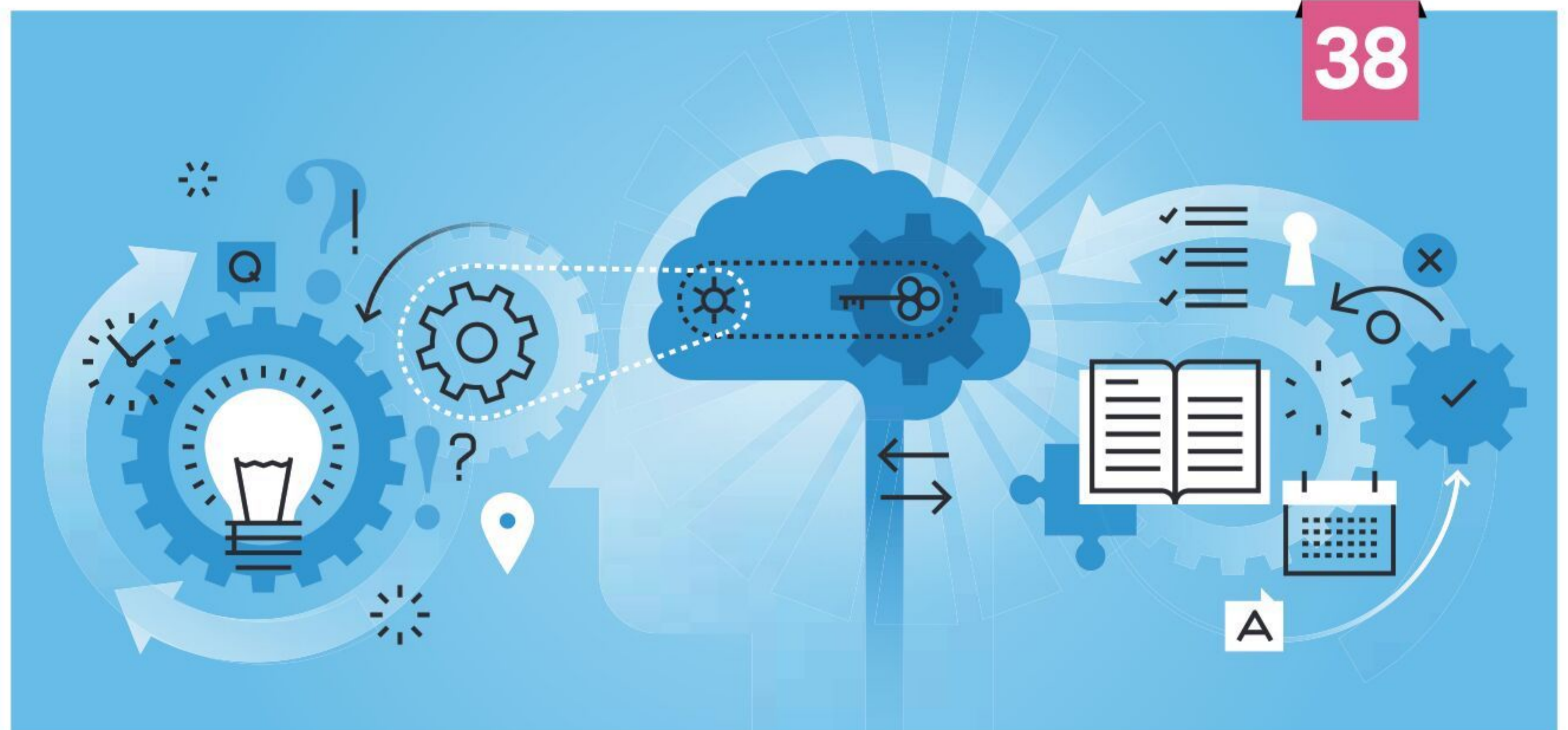
88 R Series: Regression

ADMIN

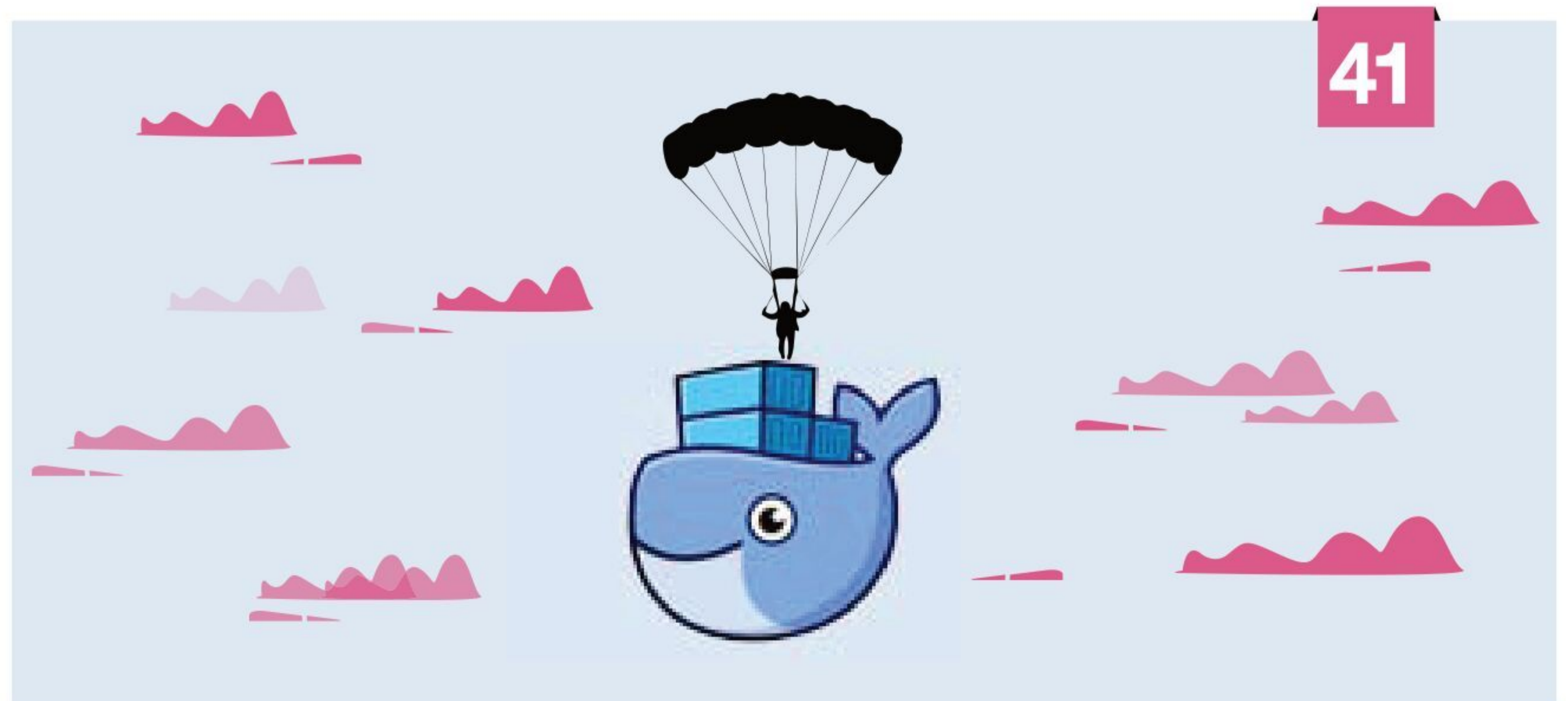
93 Integrating Network Function Virtualization with the DevOps Pipeline: Distributed Systems



Quantum Computing: Coming Faster Than You Think



Careers: The Best Way to Learn Container and Cloud Native Technologies



Getting Started with Docker

REGULAR FEATURES

07 FossBytes

Wanna Support Open Source For You? Subscribe to the Magazine— so that we can keep promoting Open Source.

AMAZING OFFER

Pay for 6 Issues Get 12!

(Buy One Get One Free!)



WORLD'S LEADING PUBLICATION ON OPEN SOURCE

TEAR OUT ORDER FORM TEAR OUT

Please
Tick (✓)
Your Choice

**Pay for 12 Issues
Get 24 Issues**
and save 50%
(Buy 12 Issues Get 12 Issues Free!)
₹1440

**Pay for 24 Issues
Get 48 Issues**
and save 50%
(Buy 24 Issues Get 24 Issue Free!)
₹2880

**Pay for 36 Issues
Get 72 Issues**
and save 50%
(Buy 36 Issues Get 36 Issue Free!)
₹4320

To subscribe online, visit
<https://tinyurl.com/y5kuv4la>

OR

SCAN
THIS
CODE



Name _____ Organisation _____ Mailing Address _____

City _____

Pin Code _____ State _____ Phone No. _____ Email _____

Subscription No. (for existing subscribers only) _____ I would like to subscribe to the above (✓)marked **Open Source For You** magazine starting with the next issue. Please

find enclosed a sum of Rs _____ by DD/MO/crossed cheque bearing the No. _____ dt. _____ in favour of **EFY Enterprises Pvt Ltd**, payable at Delhi. (Please add Rs 50 on non-metro cheque)

Please mark one (nearest) relating to your subscription: Indian Company MNC R&D organisation Engineering institute College/School Any other (specify): _____

Send this filled-in form or its photocopy to : **EFY Enterprises Pvt Ltd**, D-87/1 Okhla Industrial Area, Phase 1, New Delhi 110 020 | Ph: **011-40596600** | e-mail: **support@efy.in**

Terms:- # These rates are applicable for new subscribers as well as renewal by existing subscribers. # Can access ezine till your subscription is active # The rates are valid for subscribers within India only. # Please allow 4-6 weeks for processing of your subscription. # The subscription copies will be dispatched through ordinary post only # Subscription Agents will not get agency commission against this scheme # Disputes, if any, are subject to exclusive jurisdiction of competent courts and forums in Delhi/New Delhi only. * Replacement will be made if intimation of damaged / non-receipt of copies is received within 30 days of its publication ** After three months, if you are not satisfied with the magazine, your balance amount will be returned (Not applicable for gift offer)

EDITOR
RAHUL CHOPRA

EDITORIAL, SUBSCRIPTIONS & ADVERTISING
Delhi (HQ)
D-87/1, Okhla Industrial Area, Phase I, New Delhi 110020
Phone: +91-9811155335
E-mail: info@efy.in

MISSING ISSUES
Phone: +91-9811155335
E-mail: support@efy.in

BACK ISSUES
Phone: +91-9811155335
E-mail: support@efy.in

NEWSSTAND DISTRIBUTION
Phone: +91-9811155335
E-mail: efycirc@efy.in

ADVERTISEMENTS
NEW DELHI (HEAD OFFICE)
Phone: +91-9811155335
E-mail: efyenq@efy.in

MUMBAI
E-mail: rmowest@efy.in

BENGALURU
E-mail: rmosouth@efy.in

CHINA
Worldwide Focus Media
E-mail: china@efy.in

GERMANY
pms Plantenberg Media Service GmbH
E-mail: germany@efy.in

JAPAN
Tandem Inc.
E-mail: japan@efy.in

TAIWAN
J.K. Media
E-mail: taiwan@efy.in

UNITED KINGDOM
ASA Media
E-mail: uk@efy.in

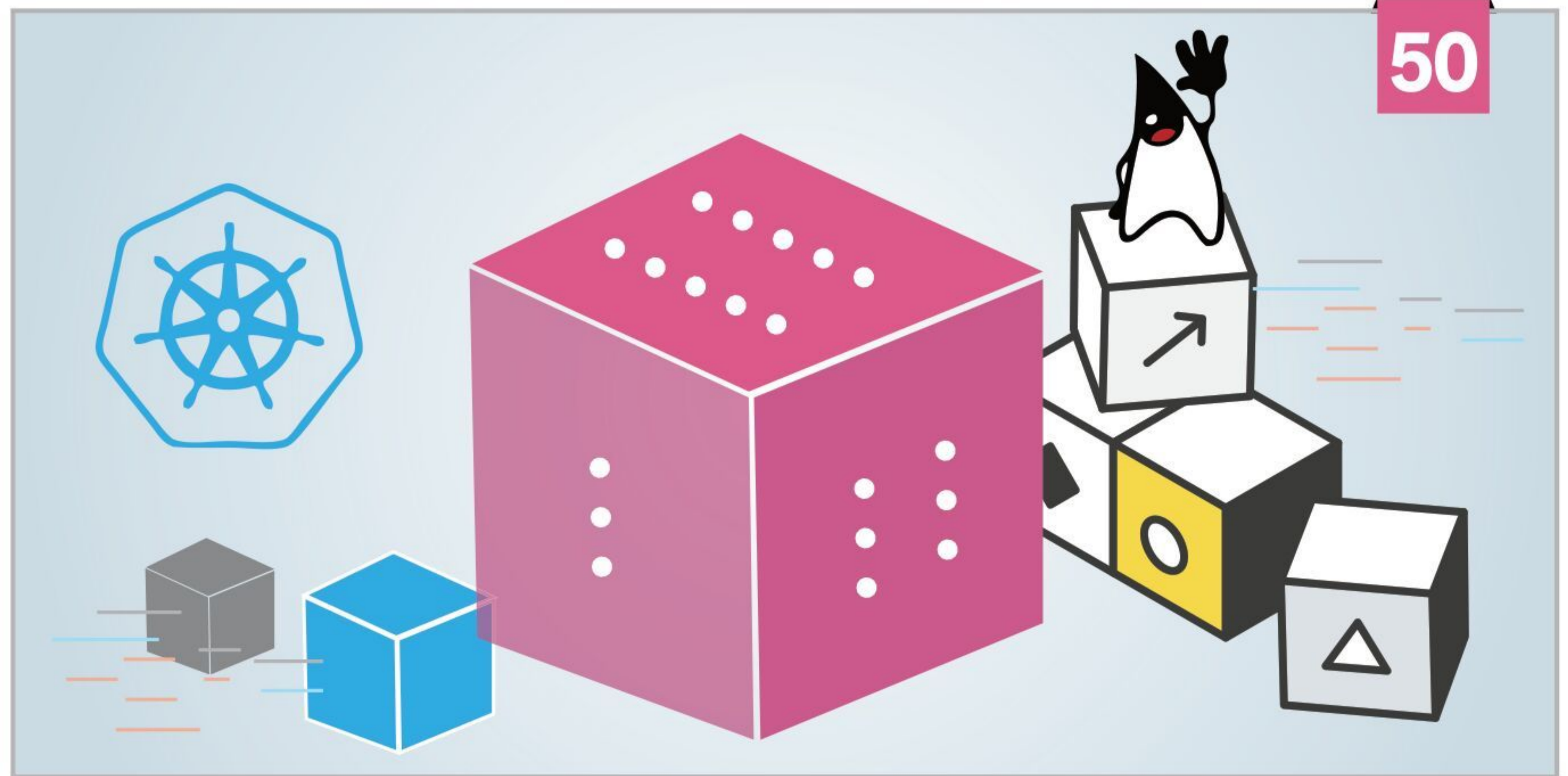
UNITED STATES
E & Tech Media
E-mail: usa@efy.in

Printed, published and owned by Ramesh Chopra. Printed at Tara Art Printers Pvt Ltd, A-46,47, Sec-5, Noida, on 28th of the previous month, and published from D-87/1, Okhla Industrial Area, Phase I, New Delhi 110020. Copyright © 2021. All articles in this issue, except for interviews, verbatim quotes, or unless otherwise explicitly mentioned, will be released under Creative Commons Attribution-NonCommercial 3.0 Unported License a month after the date of publication. Refer to <http://creativecommons.org/licenses/by-nc/3.0/> for a copy of the licence. Although every effort is made to ensure accuracy, no responsibility whatsoever is taken for any loss due to publishing errors. Articles that cannot be used are returned to the authors if accompanied by a self-addressed and sufficiently stamped envelope. But no responsibility is taken for any loss or delay in returning the material. Disputes, if any, will be settled in a New Delhi court only.

SUBSCRIPTION RATES			
Year	Newstand Price (₹)	You Pay (₹)	Overseas
Five	7200	4320	—
Three	4320	3030	—
One	1440	1150	US\$ 120

Kindly add ₹ 50/- for outside Delhi cheques.
Please send payments only in favour of EFY Enterprises Pvt Ltd.
Non-receipt of copies may be reported to support@efy.in—do mention your subscription number.

CONTENTS



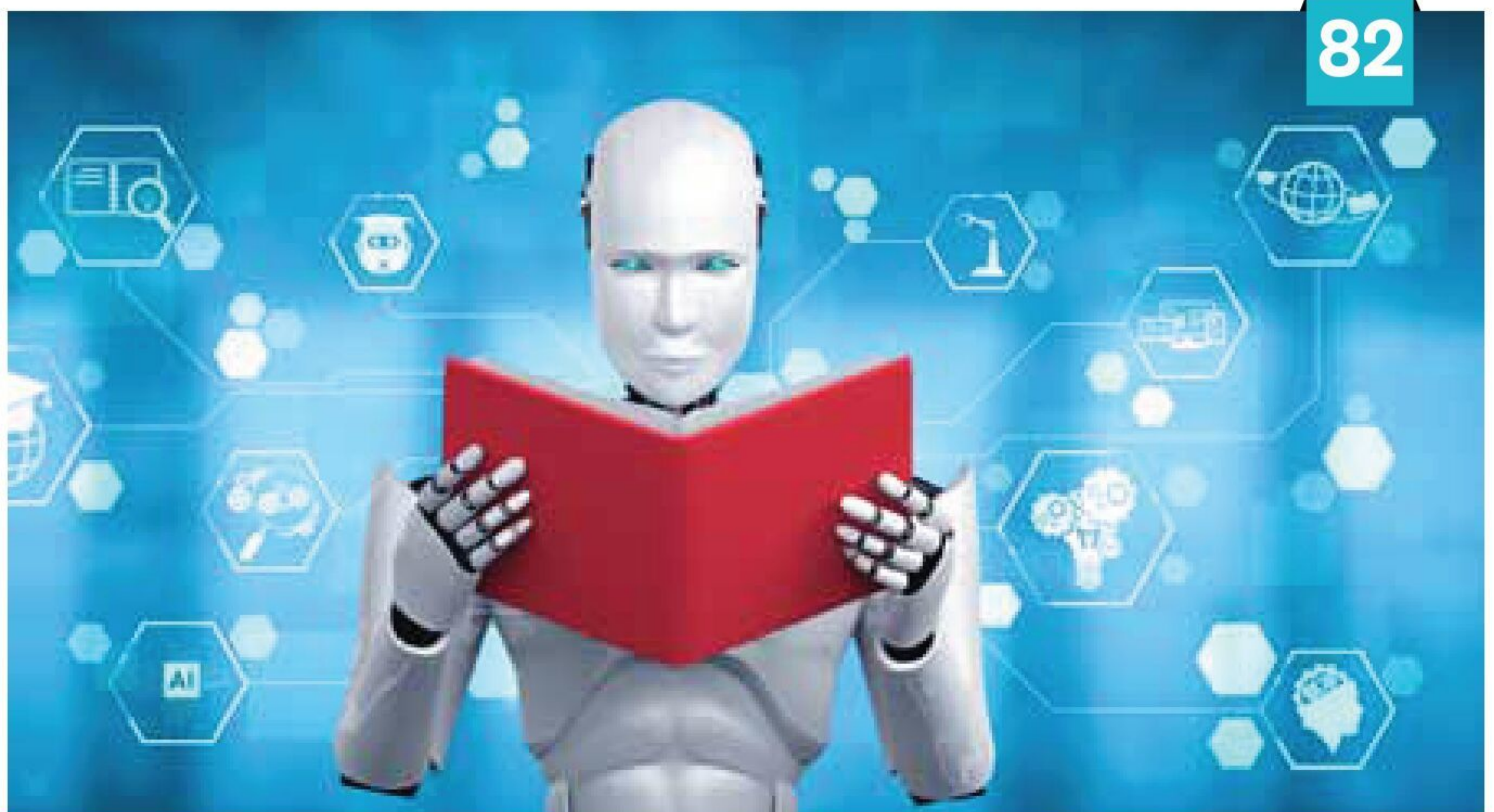
50

Deploying a Java App in Kubernetes



62

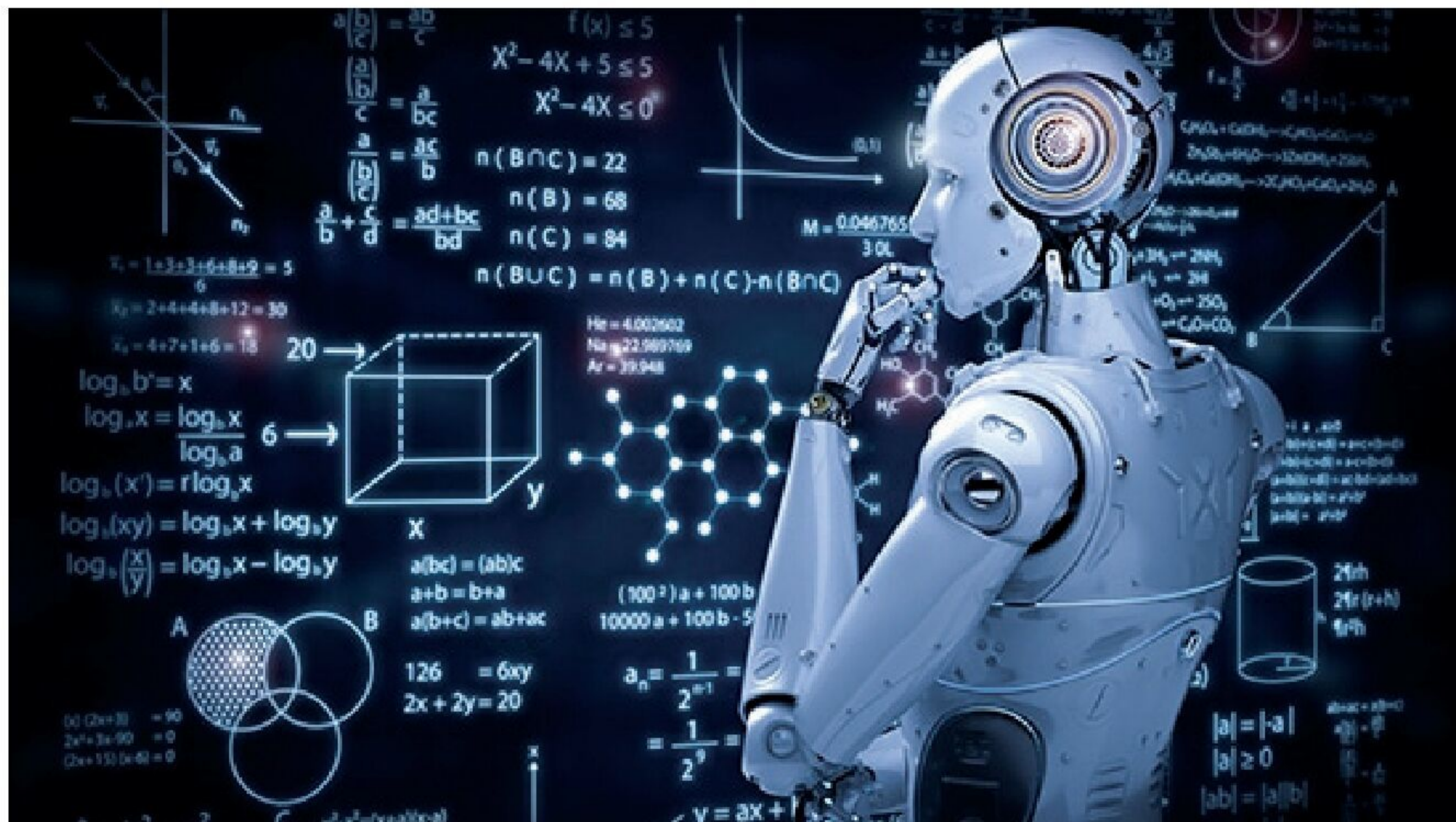
Free Tools and Libraries that Help to Keep Docker and Containers Secure



82

An Introduction to Scikit-learn and Our First Trained Model

RISELab's SkyPilot helps optimise cloud costs



Stability AI picks AWS as its go-to cloud provider

The open source artificial intelligence (AI) company Stability AI has chosen Amazon Web Services, Inc. (AWS), an Amazon.com, Inc. company, as its preferred cloud provider to build and scale its AI models for the creation of image, language, audio, video, and 3D content. Stability AI accelerates its work on open source generative AI models by utilising Amazon SageMaker (AWS's end-to-end machine learning service), as well as AWS's established compute and storage infrastructure. Additionally, Stability AI will work with AWS to make its open source tools and models accessible to worldwide startups, academics, and businesses.

Generative AI models from Stability AI can generate text, images, music, video, code, and more from straightforward language instructions. Models that are flexible to a range of tasks in areas like language, picture, audio, and video are known as generative AI or fundamental models. These models need a high-performance computing cluster with thousands of GPUs or AWS Trainium chips, extensive knowledge, and months of training. Stable Diffusion Version 2.0, which includes open source models for cutting-edge picture synthesis and enables users to produce new images from user-supplied text and image inputs, was recently released by the company.

When compared to self-managed machine learning infrastructure, Stability AI will save 58 per cent on training time and costs by using Amazon SageMaker to develop and test machine learning models. The business is leveraging Amazon Elastic Compute Cloud (Amazon EC2) to set up one of the biggest clusters of machine learning instances to run its open source models on AWS, making it simpler for developers to access those models.

SkyPilot, an open source framework for running machine learning workloads on major cloud providers through a uniform interface, has been released by a group of researchers at the RISELab at UC Berkeley. The project's main objective is cost optimisation, which involves automatically locating the availability zone, area, and provider that is most affordable for the desired resources.

The system automatically decides which AWS, Azure, and Google Cloud regions have the resources (CPU/GPU/TPU) needed to run the project, as well as which are the most cheap. SkyPilot then completes three primary tasks: it controls job queueing and execution, synchronises user code and files to the destination, and provisions the cluster with automatic failover to other sites if there are capacity or quota difficulties.

SkyPilot is not RISELab's first open source initiative aimed at reducing cloud costs. The research centre earlier released SkyPlane, which speeds up and lowers the cost of transferring huge data sets between cloud providers.

Developing multi-cloud applications, utilising best-in-class hardware, and expanding the accessibility of limited resources like top-tier NVIDIA V100 or A100 GPUs, are a few advantages of SkyPilot.

The framework has two features: Managed Spot, a way to use less expensive spot instances with automatic pre-emption recovery, and Autostop, which helps automatically clear up unused clusters.

ClamAV finally releases version 1.0.0

Despite having a tiny changelog, ClamAV recently published its most current version, which can be regarded as historically significant. The achievement is ClamAV's arrival at version 1.0.0, which was released six months after the 20th anniversary of the software.

On May 8, 2002, ClamAV's first version (0.10) was released by the program's original developer, Tomasz Kojm. Following the previously published version 0.105.1, ClamAV 1.0.0 adds a useful new capability that allows users to scan and decrypt read-only, OLE2-based.xls (Excel) files that are password-protected by default. A revamped implementation of the all-match functionality, a better scanning function in archives, a remedy for compiler warnings, and numerous other bug fixes

Ransomware toolkit Cryptonite's variant turns into a data wiper

Due to its “poor design and programming,” a variant of the open source Cryptonite ransomware toolkit has been seen in the wild with data wiper capabilities. The original source code and its branches were later removed. Cryptonite was previously distributed for free through a GitHub repository by an individual going by the moniker of CYBERDEVILZ. Written in Python, it uses the Fernet module of the cryptography package to encrypt files with the ‘.cryptn8’ extension.

A fresh sample examined by Fortinet FortiGuard Labs was discovered to encrypt files and lock them without allowing for their decryption, basically acting as a harmful data wiper.

The program’s exception-throwing behaviour also prevents operators from ever receiving the ‘key’ that was used to encrypt the files, denying them access to their data. The discoveries are set against a backdrop of a changing ransomware landscape where wipers disguising themselves as malware that encrypt files are being used more frequently to destroy data without allowing for decryption.



“The problem with this flaw is that due to the design simplicity of the ransomware if the program crashes — or is even closed — there is no way to recover the encrypted files,” Fortinet researcher Gergely Revay said.



open source operating systems. Even though ClamAV isn’t the most widely used anti-virus program for any Microsoft operating system, it is now compatible with Windows as of version 0.97.5.

ClamAV has a number of sophisticated and complex anti-virus features, including multi-threaded parallel scans, built-in support for all common mail file formats, Elf (Linux) executables, and popular document formats, as well as the ability to monitor specific folders and directories for changes.

Since ClamAV is an open source project run by volunteers with a small team of paid engineers, it is difficult to compare it to anti-virus software sold in stores.

CI Fuzz CLI checks Java programs for vulnerabilities

Java developers can now simply integrate fuzz testing into their current JUnit setup to uncover functional defects and security vulnerabilities at scale with Code Intelligence’s open source CI Fuzz CLI tool.



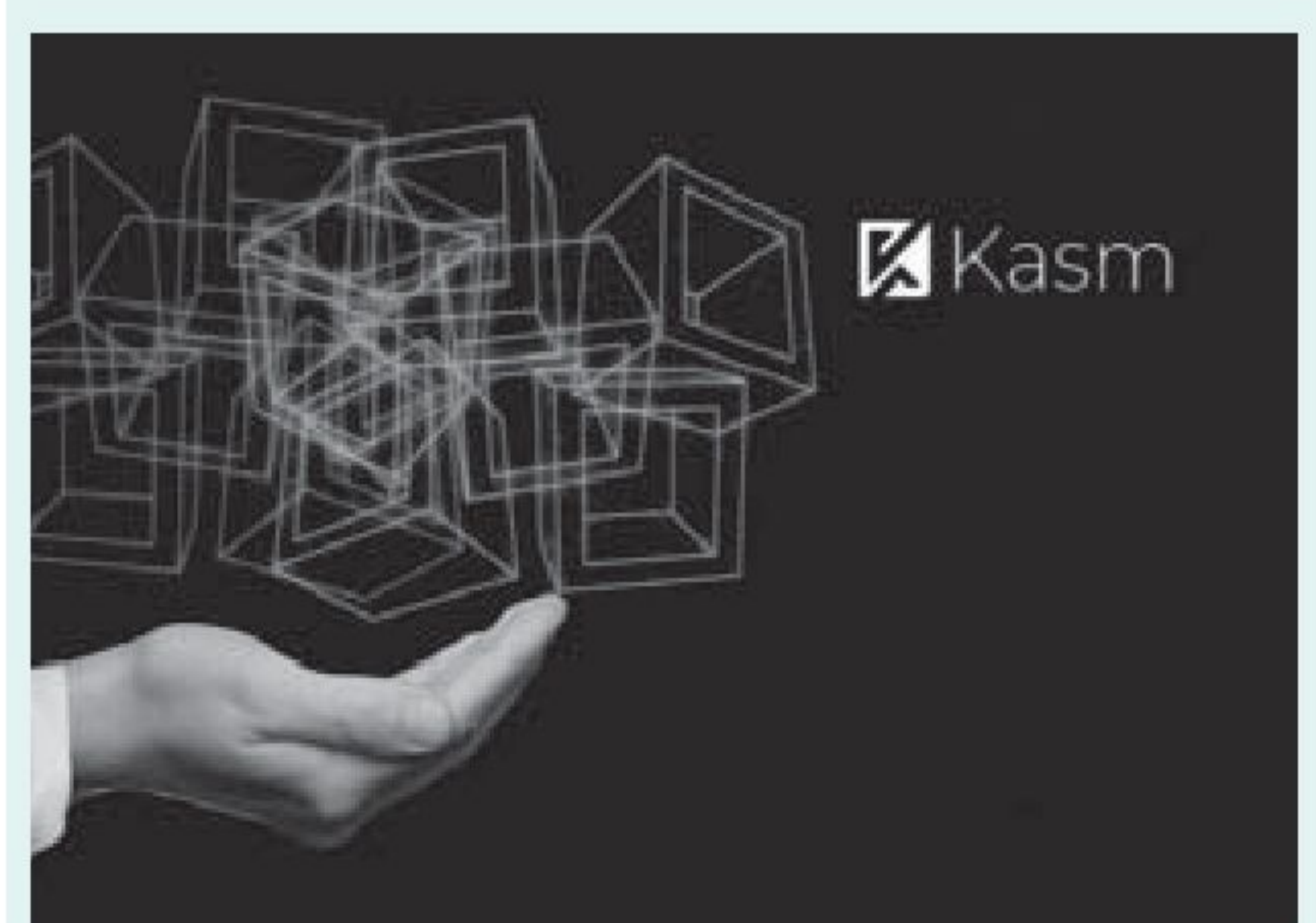
The GitHub-hosted CI Fuzz CLI uses evolutionary techniques, automated instrumentation, and millions of unique inputs to test Java programs for unexpected behaviours that could result in crashes, denial-of-service attacks, or zero-day exploits. The open source community is becoming more and more interested in fuzz testing, which can be thought of as a complimentary method to unit testing. Fuzz testing was used by Google’s Open-Source-Security (OSS) team to find more than 40,500 problems in 650 open source projects. The majority of developers outside the OSS and security communities are still learning about fuzz testing, though.

KasmVNC v1 is now available

Kasm Technologies has announced the release of KasmVNC v1. KasmVNC streams desktop, application, and web containers to the browser. The VNC server is now accessible with the release of KasmVNC v1, which is the first official, non-beta version.

KasmVNC was developed in conjunction with VirtualGL and Arm to support the next generation of streaming services. In order to accommodate contemporary technology and improve security, KasmVNC deviates significantly from existing VNC flavours by departing from the RFB specification.

Web based and agentless, KasmVNC is designed for remote desktops that are web-native and have a web server and sockets. The new version will improve streaming performance with a 150%+ frame rate boost over earlier versions. It also comes with enhanced security with HTTPS transport. Quite OK Image or QOI compression is used in order to ensure complete image accuracy while streaming at the fastest frame rate.



“KasmVNC v1 represents the culmination of a number of performance, security, and usability enhancements,” said Matt McClaskey, Kasm Technologies CTO. “These features, along with additional platform support and the documentation-as-code project, make KasmVNC the most advanced VNC server available and ready to take on enterprise environments.”

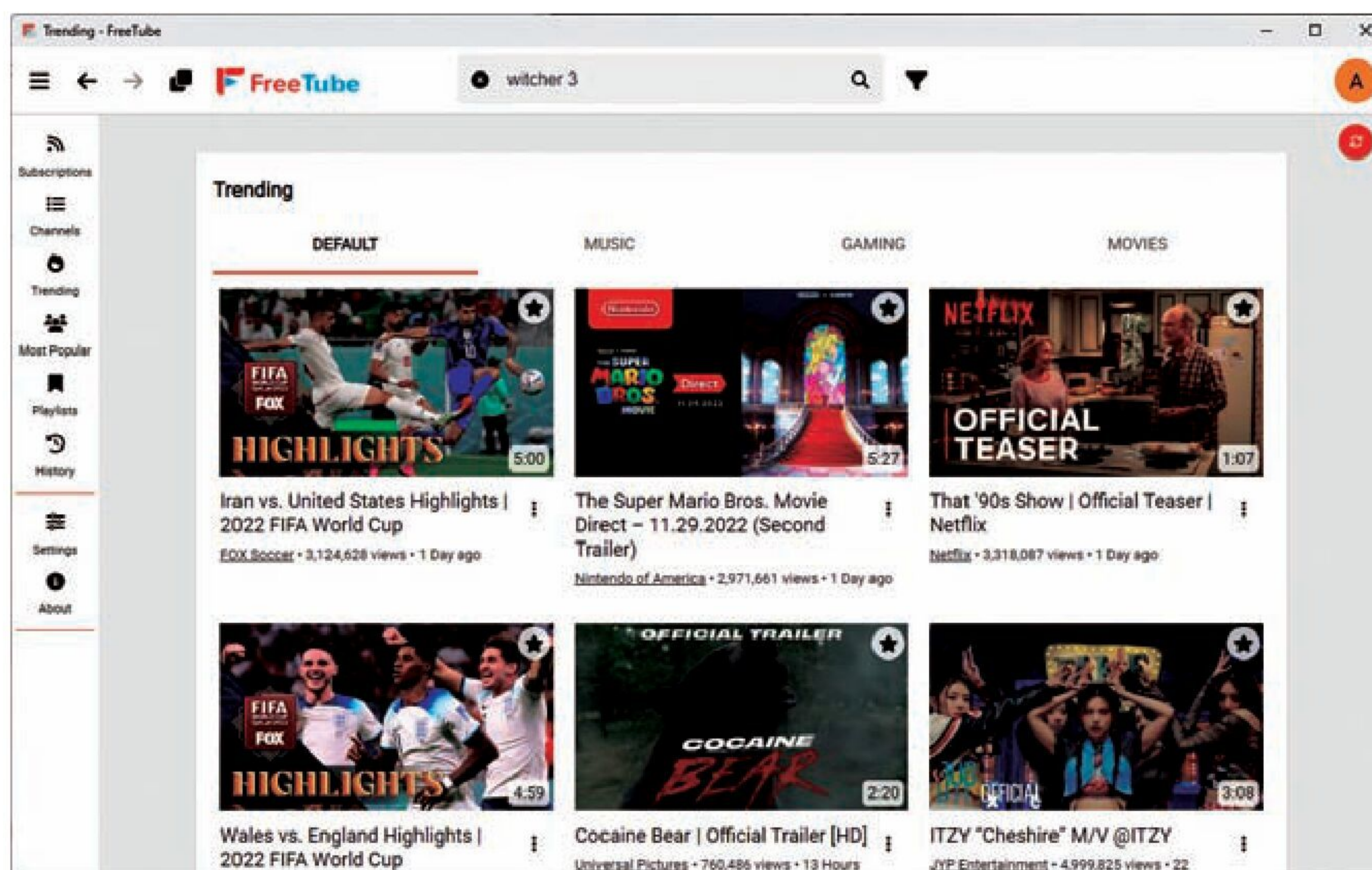
According to a recent study among Go developers, less than 12 per cent of all participants use fuzz testing at work, with implementation issues and a lack of knowledge among the main barriers to adoption. With its new open source tool, Code Intelligence hopes to address these issues by enabling all developers to use fuzz testing straight from their command line or IDE. CI Fuzz CLI offers continuous application security testing directly in the CI/CD workflow by adding additional fuzzing features for Java. This is particularly helpful for businesses that want to establish a robust DevSecOps pipeline and offer cloud based goods and services.

“With the CI Fuzz CLI, Java developers can now improve the overall security and robustness of their applications with confidence and ease. It takes just three commands to set up and run a fuzz test. The tool comes with ready-to-use integrations for Maven, Gradle and Bazel. With a JUnit setup in place, developers can even run fuzz tests directly from their IDE,” said Werner Krahe, product director at Code Intelligence.

FreeTube, an open source private YouTube client

FreeTube is an open source, cross-platform YouTube client that offers increased privacy and functionality. The free program is compatible with Windows, Linux, and Mac computers. Users of Windows can download a portable version and run it without installing it.

FreeTube is fundamentally built on Electron, which not everyone likes. For those who don't mind this, a well-designed YouTube client with more functionality than the original is available without any advertisements. The main benefits of FreeTube are an ad-free experience, lack of tracking, the ability to follow channels and publishers without creating an account, and a ton of customisability choices.



By default, the program launches with a blank screen while loading subscriptions, which are probably empty the first time. Without a Google account, subscriptions can be added, and they can also be imported from YouTube. Videos are displayed when you click on ‘Trending’ or ‘Most popular’, as well as when you use the search bar at the top to find specific articles.

The playback of a video begins as soon as it is clicked. The page resembles the original YouTube page a lot. The majority of the screen real estate is taken up by the video, but you may also see the description, related videos, channel information, and even comments, if you so want. Although it can be disabled,

Google's OSV Scanner helps check for vulnerabilities

A new tool from Google called 'OSV Scanner' enables developers to check for vulnerabilities in the open source software dependencies they utilise in their projects. To provide pertinent information about known security concerns affecting open source code, the scanner gets information from OSV.dev, the distributed vulnerability database for open source code that Google released in February 2021.

The use of a variety of pre-existing tools, libraries, and components by open source software developers often speeds up the development of more complicated solutions. These open source components can nevertheless have security flaws, just like any other type of code. These problems spread when added to other software projects. Tracking security concerns that develop with each build and assessing their possible effects on the program itself becomes a challenging chore for large projects that require numerous dependencies.

Given the number of packages that must be assessed from a security standpoint, and the fact that many of these dependencies have dependencies of their own, tracking vulnerabilities becomes a challenging task.

When a security upgrade is required, Google's new OSV Scanner automatically matches code across all dependencies for a given software project, including transitive dependencies, and notifies the developers. The deployed package version of the scanner leverages publicly available advisories from reputable, authoritative sources in accordance with the OSV template for vulnerability triage.

The Linux Kernel, Android, Debian, Alpine, PyPI, npm, OSS-Fuzz, and Maven are just a few of the 16 key coding ecosystems that the OSV.dev service currently supports.

autoplay is supported. Videos can be shared, saved and downloaded, or their format can be altered. You can switch between subtitles, picture-in-picture mode, different playing speeds, and different video resolutions.

Videos can be seen in FreeTube, but you can also choose to watch them on a specific external player, an invidious instance, or YouTube.

Movetodon makes it easier to find Twitter contacts on Mastodon

Movetodon, which was released recently, is quickly gaining popularity among Twitter emigrants.

Following Elon Musk's purchase of Twitter, open source Twitter competitor Mastodon experienced rapid growth, surpassing the 1 million monthly active user mark in less than a week. However, one of the many difficulties for users joining Mastodon for the first time is re-establishing their Twitter network of friends and followers that they had amassed over the years. Movetodon attempts to ease the transfer from Twitter to Mastodon by making it simple for users to locate and follow their Twitter connections.



You'll be able to follow certain users by clicking a button after the list generation is finished, or you can decide to follow everyone at once. To make new accounts stand out at the top of the list, you can also order the list by the date that users joined Mastodon. The creator claims that his application analyses pertinent variables on Twitter, such as a person's bio, user name, location, URL, or pinned post, where individuals frequently reveal information about their Mastodon accounts.

Movetodon doesn't store the data it accesses on its server. For instance, Movetodon cannot publish or follow on Twitter; it can only consume information.

There are other tools too, such as Debirdify, Twitodon, and Fedifinder, for finding Twitter pals on Mastodon. They are useful if you want to simply import all your Twitter friends who have Mastodon accounts, but don't allow you to easily curate your network if you want a fresh start.

Kunlun Tech releases a complete set of AIGC algorithms

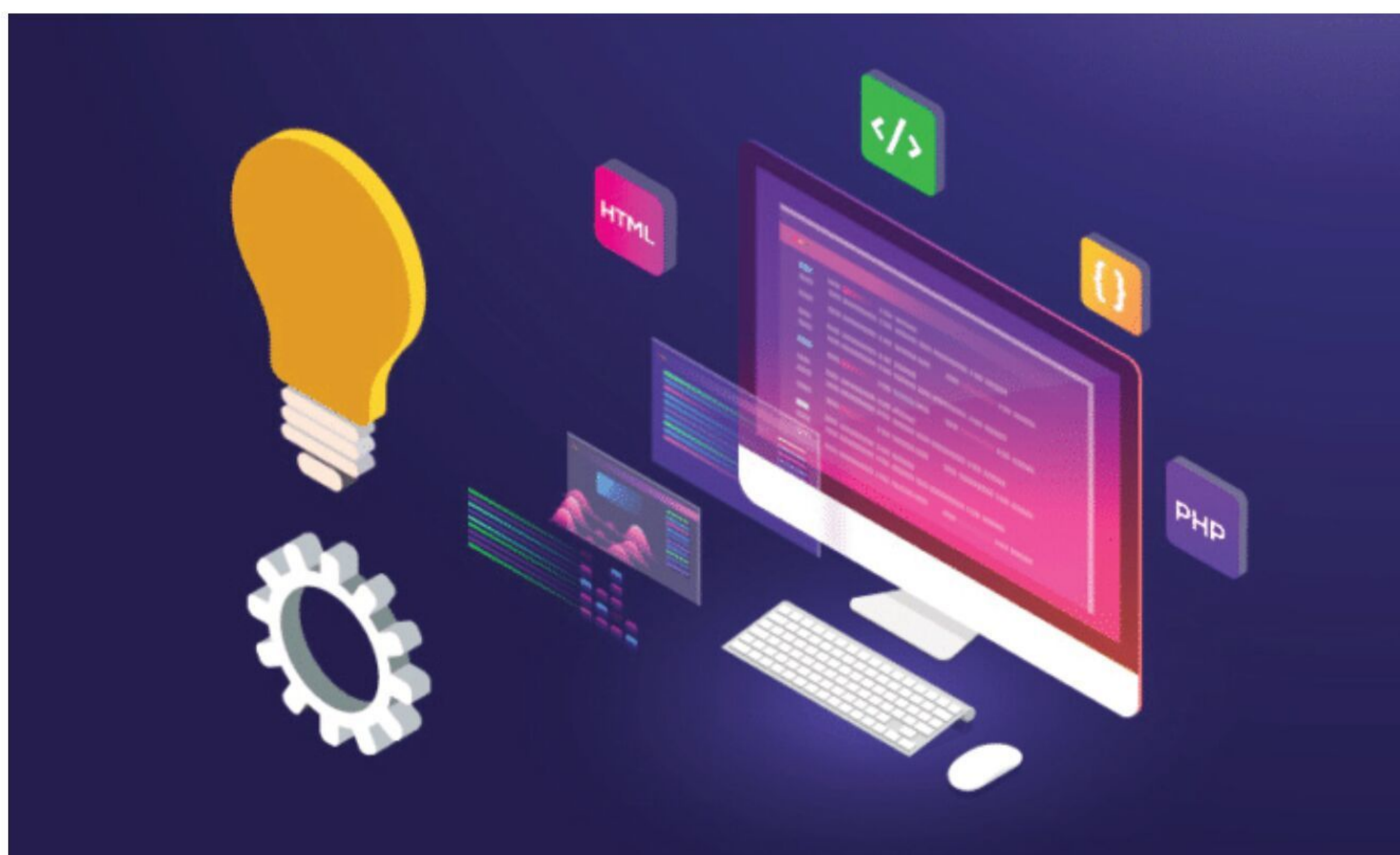
During an AIGC (AI generated content) technology conference held by Chinese company Kunlun Tech in Beijing, CEO Fang Han announced an entire series of open source AIGC algorithms and models.

The models, which cover multi-modal content generation capabilities such as image, music, text, and programming, include SkyPaint, SkyMusic, SkyText, and

SkyCode. Kunlun Tech has perhaps the most comprehensive layout in the AIGC field in China, and is the first company to dedicate itself to the AIGC open source community in that country.

SkyMusic is the first AI model for making music of a commercial calibre in China. The model adds global information restrictions and chord implicit representation. Chord accuracy reaches 92.5 per cent, significantly above the current industry high of 64.7 per cent. It also supports a global multi-song style and 31 languages, including Chinese, English, Japanese, Korean, and French. Based on the extensive collection of top-notch works available on Kunlun Tech's StarMaker platform, SkyMusic has the capacity to effectively deliver top-notch AIGC music, including songs, soundtracks, and ambient music with ISRC copyright certification.

In terms of AI text, SkyText creates 100 billion high-quality data sets for the Chinese domain, using the best open source GPT Chinese pre-training large model. This is done using a high-performance A100-GPU cluster, training 200 graphics cards over the course of four weeks, and then optimising the model over the course of two weeks to produce a 10 billion parametric GPT-3 generating model that includes continuation, dialogue, Chinese-English translation, and content style generation.



SkyCode's AI programming uses the first multilingual open source programming large model in the world and supports a number of popular code languages. SkyCode can produce code in more than ten programming languages, including PHP, Go, and Swift, in addition to the widely used Python, Java, C, etc, so that users of various languages can benefit from its potent code creation abilities.

Open Source Risk Report: Vulnerabilities are on the rise

The *Open Source Risk Report*, released recently by Mend, exposes the enormous risk posed by the continued rise in open source vulnerabilities and software supply chain threats. The report states that the number of open source vulnerabilities that Mend discovered and added to its vulnerability database in the first nine months of 2022 was 33 per cent higher than the same period in 2021, reflecting both the acceleration of vulnerabilities and the growth in the number of published open source packages. This expanding vulnerability is a significant problem as organisations continue to rely heavily on their applications for success.

According to the report's representative sample of 1,000 North American organisations, studied from January to September 2022, just 13 per cent of vulnerabilities were patched, compared to 40 per cent for those who used

OpenShot 3.0 has new features and major enhancements

Over 1,000 fixes and improvements have been made in OpenShot 3.0, along with enhanced memory management, stability, and the introduction of intriguing new export and performance features. The non-linear video editor had not received an update since 2021.

In a blog post describing the modifications, OpenShot's developer Jonathan Thomas wrote that he and the team "worked tirelessly over the past 12 months to identify and fix as many stability issues as possible." Due to persistent performance concerns, those who are unfamiliar with the program might not be aware that it goes by the moniker 'OpenShut'.

Additionally, the editor's video decoding engine has been modified, which is supposed to result in "smoother" real-time video playback with "fewer freezes and stops" during preview. One of OpenShot 3.0's significant flaws is the inability to consistently preview adjustments as you make them. Of all the improvements, this one could be the most helpful to those who intend to use the app frequently. OpenShot 3.0 now supports multi-threaded video formats and codecs, such as AV1. It also refactors video caching (making incoming frames available before they are required) and adds a new 'clean' cache option.

There are a number of noteworthy timeline changes, such as enhanced snapping precision, quicker clip slicing, improved keyframe icons, distinctive colours for video effects, and transitions that "define their own course" (e.g., fade-in at the start of a clip, fade out at the end, etc).

New ransomware is using a carefully disguised payload

According to recent research by Checkmarx and Phylum, an ongoing ransomware campaign targets well-known open source packages that regularly see close to 15 million installations every week and conceals its payload in an unusual method.

The campaign, which includes embedded malware, targets the well-known ‘requests’ package on PyPi and the ‘discord.js’ package on NPM, according to a blog post by Checkmarx researchers. When the ransomware is run, it encrypts the victim’s computer data and demands US\$ 100 in cryptocurrency to decrypt them.

Alik Koldobsky, a security researcher at Checkmarx, told *SC Media* that the payload is hidden in several key locations and only executes when the victims use the actual functions of the packages, unlike most open source attacks where malicious packages are run immediately after installation. This makes the campaign difficult to detect by many security scanners.

Because the malware payload is compatible with several different operating systems, the campaign can target a larger population. Although a complete attribution has not yet been made, researchers have found evidence that suggests the attacker is Russian -- the Telegram user account linked to the attack has a Russian phone number, and the attacker directly communicates with researchers in Russian.

The perpetrator’s account is still able to distribute potentially dangerous packages on NPM and PyPi, where software supply chain assaults are common, even after Checkmarx reported the attacks. According to researchers, they will keep an eye out for any new activity.

contemporary application security best practices. Today, 70 to 90 per cent of applications employ open source code, making more businesses susceptible to attacks as threat actors take advantage of the repair gap.

Even when businesses patch tens of thousands of vulnerabilities each month, it takes contemporary remediation best practices to handle the constant stream of new vulnerabilities discovered and stop the vulnerability backlog from increasing. Open source software availability has grown by an estimated 25 per cent, yet the rise of open source vulnerabilities has outpaced that growth. The use of prioritisation and remediation technologies, as well as routine application security



scanning, is crucial because apps are the backbone of the global economy.

“As security debt continues to rise, it’s crucial to find a way to prioritise the vulnerabilities that pose the highest risk to avoid falling victim to an attack,” said Jeffrey Martin, VP product management at Mend. “Using remediation tools that can assess and prioritise the vulnerabilities that can most heavily impact systems is an important element to managing security debt. Organisations should not just pay attention to severity details though; to ensure effective prioritisation and remediation, they need to also look at the exploitation context of flaws on their own and in conjunction with others.”

ActiveState Artifact Repository is available for free for a limited period

ActiveState is providing the ActiveState Artifact Repository without charge in all ActiveState Platform tiers (including its well-liked Free Tier) for a limited period. In September 2022, the ActiveState Artifact Repository was made available, making it the first tool in the market to let businesses use ActiveState’s secure build service to safely generate Python open source dependencies from source code. A closed-loop environment is thus created, maximising supply chain security, whereby all produced code is immediately stored for distribution in the company’s own private ActiveState Artifact Repository.

Since its introduction, the ActiveState Artifact Repository has proven useful to ActiveState customers and interfaces with a number of other premier secure supply chain capabilities that are pre-integrated into the ActiveState Platform, such as curated collection of examined artefacts, functionality for resolving vulnerabilities, a certification programme for open source products, and a software bill of materials (SBOM).

ActiveState makes it simpler than ever for enterprises to put security at the centre of their open source supply chain by offering the ActiveState Artifact Repository free of charge for all tiers of service and coupling it with other crucial

ActiveState

open source security capabilities.

Cybersecurity organisations in the UK and US have both recently provided recommendations to help firms deal with the increased

threat as a result of the rise in supply chain assaults in recent years. The National Institute of Standards and Technology (NIST) released comprehensive guidelines in September 2022 to assist software businesses in fending off supply chain threats. The National Cyber Security Centre (NCSC) of the United Kingdom released similar advice in October 2022. It is anticipated that this advice will soon become the industry norm.

Drupal introduces Drupal 10

Drupal stands out for being flexible, and modularity is one of its guiding principles. Users will have access to even more fantastic tools with Drupal 10 to help create the flexible, structured content that dynamic online experiences demand. With a revamped backend appearance and a platform upgrade that the company says is future-proof, this most recent edition elevates Drupal to a new level. The new edition offers robust new features and a better user interface.



Developers have been working to improve Drupal for the past two-and-a-half years. The Claro administration theme, Olivero front-end theme, CKEditor 5, and other new features support Drupal's vision for the web's future.

Drupal 10 is based on Symfony 6.2, the most recent release. Additionally, the least-used parts of the core system are eliminated, streamlining the system and enhancing its underlying strengths. With nearly all code changes automated and three times as many contributed extensions available at launch time as with the previous major update, the upgrade tools and procedures are the same as with the previous major release.

Endor Labs' report highlights the risks of using unmonitored open source software to develop applications

'The State Of Dependency Management', a report from Endor Labs, a startup committed to securing open source software reuse in application development, provides insight into the pervasive but frequently unmonitored use of existing open source software in application development and the risks associated with this practice.



For instance, according to the report, 95 per cent of all vulnerabilities are discovered in open source code packages with transitive dependencies that are not explicitly chosen by developers but instead are unintentionally incorporated into projects. This is the first report from Station 9, a research platform created by Endor Labs that gathers experts from many fields throughout the world.

The issue isn't simply the broad use of open source code in new apps; rather, it's the fact that developers only choose a small sample of these software requirements. The remaining dependencies are transitive or indirect ones that the codebase automatically incorporates. This creates the conditions for important, possible vulnerabilities that could have an equal impact on development and security.

For more news, visit
www.opensourceforu.com



Quantum Computing: Coming Faster Than You Think

From cybersecurity to drug discovery, there are several challenges that quantum computing can tackle better than classical systems. With improving hardware, full-stack solutions, better algorithms, simulators, and cloud offerings, the technology is at present in ‘democratisation’ phase. Perhaps it is time to make your quantum plans too.

IBM Quantum Lab at the Thomas J. Watson Research Center (Credit: Connie Zhou for IBM)

“G iven a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?” First formulated in 1930, this mathematical problem, known as the ‘travelling salesman problem,’ is one of the hardest to solve, yet one of the most essential optimisation issues for any logistics provider.

“Quantum computing has the potential to blow classical computing out of the water for certain types of complicated problems. For example, constrained optimisation challenges like the famous travelling salesman problem can have trillions of equations and variables. The most complex of these problems will simply be unfeasible for classical computers to tackle, but quantum computers will be able to solve them

in seconds. And these aren’t just laboratory problems; constrained optimisation challenges heavily impact industries like supply chain and retail, cybersecurity, life sciences and pharmaceutical, and government and applied research,” says Robert Liscouski, CEO of Quantum Computing Inc. (QCI).

Another great example involves autonomous vehicles. QCI recently solved a 3854-variable sensor placement problem for BMW using only quantum computing—and they did it in six minutes. This is a clear example of quantum computing’s ability to solve actionable, real-world problems that would take classical computers a much longer time to solve.

“To clarify, sensor placement is extremely challenging because of all the variables that must be considered, including wind resistance, weight



balancing, chassis design, potential obstructions, and more. Our machine found a way to provide 96% vehicle coverage using only 15 sensors. So, when you ask about potential, we're talking about saving autonomous vehicle designers a tremendous amount of money. This is compared to the solutions provided by other quantum and classical computers that suggested using well over 100 sensors," explains Liscouski.

Mercedes-Benz, which hopes for all its vehicles to be carbon-neutral by 2039, is working with IBM Quantum to simulate the chemical reactions in batteries more accurately. Basically, they are trying to understand what happens at a molecular level inside the battery, while it is working. This is a task that is practically impossible for classical supercomputers, as it involves a tremendous number of electron interactions, each influencing the other in complex ways. No wonder, Dr Jeannette (Jamie) Garcia, Senior Manager of Quantum Algorithms, Applications and Theory at IBM, remarks that, "It is chaos in there!"

"Most industries have their own big, intractable challenges. And quantum computers offer potential solutions. For instance, quantum computing can shed light on processes of molecular and chemical interactions," says L. Venkata Subramaniam, Senior Manager - AI Science and IBM Quantum Ambassador, IBM Research India. Mitsubishi Chemical, Keio University and IBM are working together to model and study the complex mechanism for lithium superoxide rearrangement, a key chemical step in lithium-oxygen batteries. Using a quantum computing setup enables them to more accurately measure the reactant energy used to produce electricity in a Li-air battery.

So, is quantum computing only for researchers and designers? No, every large business house can use it to optimise its running. "Quantum computing is enabling ExxonMobil to solve several computationally-



Quantum computer mixing chamber (Credit: IBM Research)

challenging problems such as optimising natural gas supply and the discovery of new materials for efficient carbon capture," says Subramaniam.

Efficient shipping of liquid natural gas (LNG) is extremely critical, because people could run out of power without it! The global LNG industry involves thousands of voyages across the world. Optimising these, at the most basic level, would involve millions of discrete decisions. If you take into consideration uncertainties like bad weather and market fluctuation, it could run into even trillions.

According to Dr Vijay Swarup, Director of Technology at ExxonMobil, this means that the number of combinations they would need to consider would be larger than the number of atoms in the entire universe. ExxonMobil is working on modelling maritime inventory routing on IBM's quantum setup.

Think quantum

"We tend to think of problems and solutions in classical ways, for which the existing classical computational ecosystem is fast and powerful enough. Most of the standard/classical problems can be adequately addressed within the existing infrastructure. However, there exists a certain class of problems that quantum computers can solve in more effective ways," explains Rahul Mahajan, CTO, Nagarro.

"Drug discovery, security, and search are a few domains where we foresee some interesting applications

using quantum computers. Given a sufficiently large quantum computer, there exist a few algorithms like Grover's and Shor's, which can be deployed on quantum machines to execute the solution faster. For example, a class of RSA algorithms use factorisation based encryption, leveraging quantum parallelism in qubits registers, and simultaneous evaluation paths can be traversed," he says.

"Quantum computing makes different types of seemingly intractable optimisation problems, like routing of delivery vehicles, more manageable. Another class of problems comes from chemical reactions. These are by nature quantum mechanical. The best way to understand them is via simulations on a quantum computer. With enough qubits, we could tackle problems in electric battery design, find new molecules, and aid drug discovery. The societal benefits that would follow are immense," says Professor Anil Prabhakar, Department of Electrical Engineering, IIT Madras.

In the last few years, there have been significant developments in this space—mergers, alliances, stock launches and public-private partnerships; better algorithms, full-stack solutions, hardware upgrades and concrete tech roadmaps; and an expanding portfolio of applications. We see people rethinking quantum deployments, often opting for quantum simulators, and hybrid systems with quantum computers working hand-in-hand with CPUs and GPUs. We see the emergence of cloud ecosystems that mask the complexity, enabling even non-quantum experts in different fields to use quantum computing to solve stubborn problems.

We do not have to wait for quantum computing to replace classical systems, to finally agree that quantum is here—because that will probably never happen. We need to understand that classical and quantum

do not actually compete—the purpose of a quantum computer is not simply to solve classical computing problems faster than a supercomputer. It is fundamentally different, and ideally suited to crunch complex problems that require calculating a large number of possible combinations.

The problems best left to quantum computers are known as bounded-error quantum polynomials (BQP), that is, problems that can be solved by a quantum computer in polynomial time, such as factoring, searching an unordered database, and solving systems of linear equations. Problems like this abound in fields like optimisation, materials science, cryptography, artificial intelligence, machine learning, sensing, telecommunications, and finance.

Many processes in nature also involve many-body quantum interactions. This makes quantum computing an essential tech to simulate natural processes like protein folding, molecular formations, photosynthesis, and superconductivity, and use this understanding to develop better nature-inspired technologies.

“Today I cannot see that there will be something like a quantum PC,” quips Professor Rainer Dumke, Principal Investigator at the Centre for Quantum Technologies, Professor at Nanyang Technological University, Singapore, and Lead for hardware at Singapore’s National Quantum Computing Hub (NQCH). “We will witness in the future that more and more data centres will have a quantum computer on the premises. To give a meaningful timeframe for this is very challenging, especially at this early stage. However, we see already that supercomputers and data centres around the globe are adopting quantum computers of the current generation.”

At the NQCH, they are working to establish a Singapore-built quantum computer, which can be accessed by users remotely. They also hope to develop an advanced quantum processor test platform, along with various components like chips, fast

Horizons of quantum growth

Horizon 1, also called now or near term, includes transactional use cases, such as credit scoring, vehicle routing, chemical design, chemistry, and drug/protein structure prediction.

Horizon 2, also called near term, follows with oil processing and shipping, refining processes, drilling, livestock, disruption management and supply chain issues, investment risk analysis, clinical trial acceleration, and optimisation of manufacturing and fabrication.

Horizon 3, also called future looking, consists of seismic imaging, consumer recommendation with financial analysis, disease risk prediction, and structural design for buildings.

(Credits: Paul Nashawaty, Senior Analyst at Enterprise Strategy Group, a division of TechTarget)

electronics, and cryoelectronics for the next generation of superconducting quantum processors.

Overcoming the hardware challenges

In an actual quantum computer, physical systems, such as the spin of an electron or the orientation of a photon, are used to store qubits. These, as you can imagine, are pretty fragile systems, and are easily affected by various factors, such as disturbances in the earth’s magnetic field, radiation from devices in the vicinity, heat, neighbouring qubits, and so on. These factors, which affect the information in a qubit, are known as ‘noise,’ and since today’s quantum computers are prone to their effects, they are known as noisy intermediate-scale quantum (NISQ) devices.

Unfortunately, even reading or accessing a qubit amounts to noise! Noise can cause the information in the qubits to change, or even fade away! This is known as decoherence—that is, the information in the qubit is no longer coherent. Noise and resulting decoherence are two of the biggest hurdles in the way of quantum computers.

Quantum computer makers try to protect the qubits from noise by using cooled silos, controlled pulses of energy, etc. So, if you look at a quantum processor, such as the one from IBM, the wafer is just about the size of the one in your laptop, but it is housed in

a hardware system the size of a car, made up of cooling systems to keep the superconducting processor at its ultra-cold operational temperature.

The task of mitigating the effect of noise is known as quantum error correction. Mostly, a lot of spare physical qubits are used to store every logical qubit, so that in case one loses its coherence you can verify with another. Researchers also use noise simulations to try and understand the nature and effect of noise, and develop noise-tolerant algorithms, which will give fairly good results despite the noise. Finnish company Algorithmiq is working to develop noise-resilient quantum algorithms for drug discovery. Their strategy involves using standard computers to ‘un-noise’ quantum computers!

“Techniques for error mitigation and error correction are likely to be the way forward on NISQ computers. There is currently a trade-off between the number of qubits we can use and the depth of the quantum circuit. The product of the two is referred to as the quantum volume. Researchers are finding ways to increase the quantum volume, and demonstrate the advantage of quantum algorithms. One such approach is hybrid quantum computing, where the computational task is shared between classical and quantum processors,” says Prof. Prabhakar.

“The research on quantum fault tolerance is ongoing, and the initial



Start your open source journey...

Multicloud Data Management

■ Data mobility ■ Data protection ■ Data observability

SODA Foundation, an open source project under The Linux Foundation, aims to foster an ecosystem of open source data management and storage software for data autonomy. SODA Foundation offers a neutral forum for cross-projects collaboration and integration and provides end users with quality end-to-end solutions.

Contribute at <https://github.com/sodafoundation>



Slack



GitHub



SODA CDM

SODA Framework Projects



TERRA

SDS
Controller



DELFIN

Storage
Monitor



STRATO

Multicloud
Data Manager



KAHU

Container
Data
Protection



COMO

Virtual
Data Lake

www.sodafoundation.io

Job Guaranteed.

You don't need a training institute. Just do it yourself.

WORK WITH OPEN SOURCE.

www.opensourceforu.com

Demand for Open Source is sky rocketing. Be it for managing IT infrastructure or development of software —Open Source experts are what organisations are seeking. Develop expertise in an Open Source stack, and watch your career graph zoom.

You don't need a training institute for this. You can do it yourself. And, *Open Source For You* can be your friend and a guide through this journey.

TO READ OUR PRINT EDITION Visit: <http://subscribe.efyindia.com>

TO READ OUR EZINE EDITION Visit: <http://ezine.efyindia.com>

WORLD'S LEADING PUBLICATION ON OPEN SOURCE

Looking for marketing solutions to engage with cutting edge techies?
Contact us at growmybiz@efy.in OR call us at +91-9811155335.



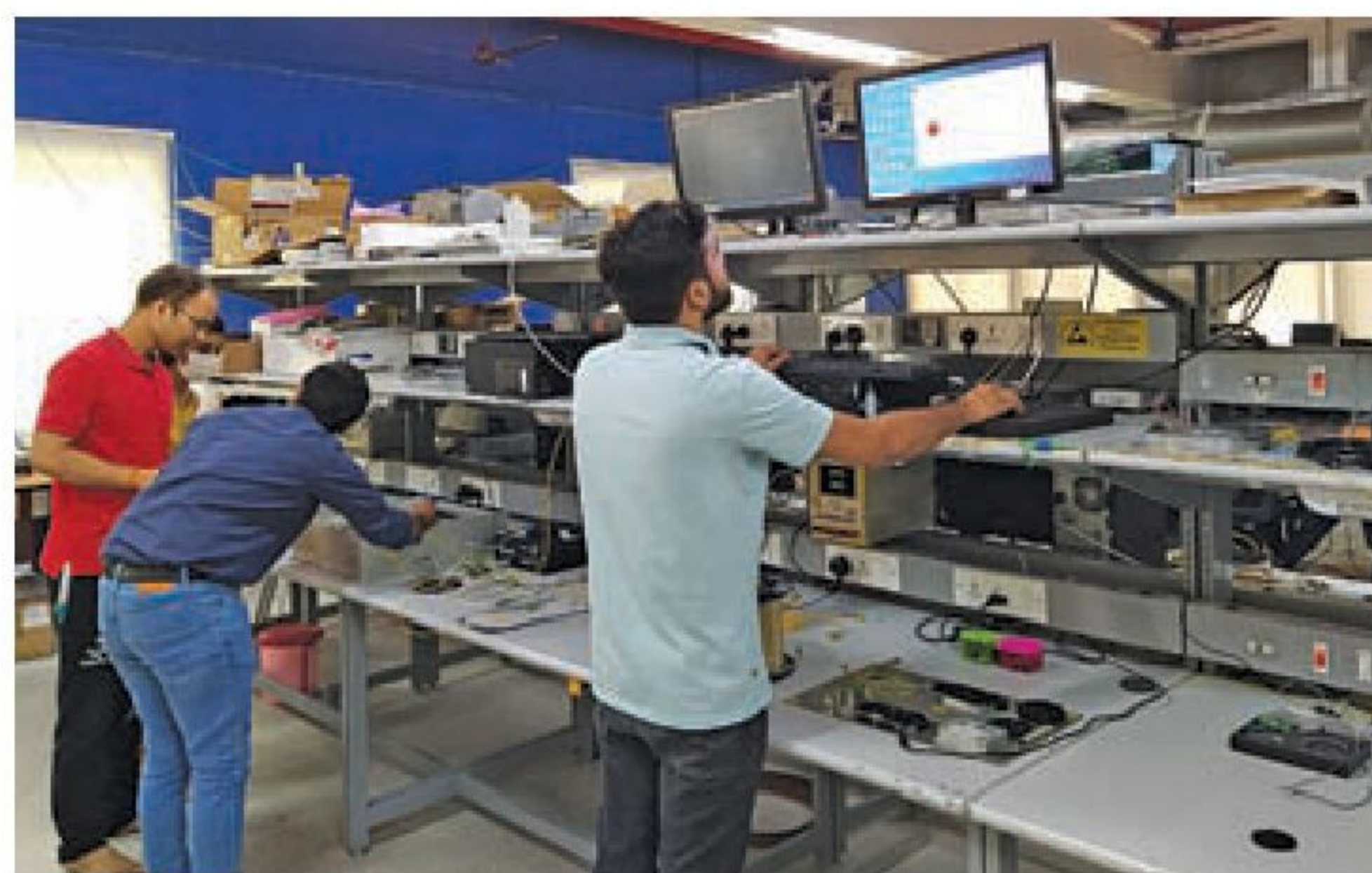
research was done by Peter Shor. The latest innovation is about generating sufficient redundancy to maintain information integrity. These redundancy methods have negatively introduced logarithmic downsizing on actual logical qubits available for compute, and as of today, we are barely in the range of a few dozen logical qubits,” explains Mahajan.

Subramaniam shares that, “At IBM, we are developing bottom-up approaches to the problem of noisy qubits and incorporating error mitigation techniques to realise this technology’s true potential. Superconducting qubit based systems have made tremendous strides in device performance, from improved coherences to lowered single- and two-qubit gate errors, and high-fidelity mid-circuit measurements and qubit resets. In the future, we plan to bring together advances in device performance, software and error correction to offer users a frictionless experience and the ability to solve tough computational problems.”

In 2021, IBM unveiled Eagle, IBM’s first quantum processor developed and deployed to contain more than 100 operational and connected qubits. “It is the first IBM quantum processor whose scale makes it impossible for a classical computer to reliably simulate,” says Subramaniam.

IBM has the 433-qubit Osprey processor slated for release later this year, and the 1121-qubit Condor processor slated for release next year. In 2024, IBM plans to introduce longer-range quantum communication between chips and create clusters of quantum processors using a long-range coupler for connecting qubit chips through a cryogenic cable of around a metre long. This will be demonstrated by linking together at least three 462-qubit processors, each called Flamingo, into a 1386-qubit system.

IBM’s next quantum processor, Kookaburra, is also slated for release in 2025. It will be a 1386-qubit multichip processor with quantum communication



Quantum Communication Lab (Credit: IIT Madras)

link support for quantum parallelisation. IBM also plans to connect three Kookaburra chips into a 4158-qubit system connected by quantum communication.

Beyond superconducting qubits

Prof. Prabhakar remarks that most people tend to think only of superconducting qubits from companies such as IBM or Google, when considering quantum computers. “There are other hardware platforms also gaining traction, like ion-traps, cold atoms, and photonic cluster states. These are very different approaches and I expect that they will yield application-specific quantum processing units (QPUs) in the near term,” he says.

QCI has developed one such system, which Liscouski describes to us. “Natural quantum states interact freely, influencing and impacting each other as they evolve and change. This natural interaction (or ‘noise’) significantly impacts the accuracy and scale of first generation NISQ computers. These vendors painstakingly create hyper-cooled, vacuum environments to house their computers, but even this doesn’t prevent them from making significant errors, losing information, and suffering from limited scale. Our photonics-based Entropy Quantum Computer (EQC), on the other hand, harnesses the fundamentals of quantum physics to overcome these limitations. It operates on open quantum systems, carefully coupling to an engineered environment, so that the quantum state is collapsed to represent a problem’s desirable solution. In

layman’s terms, where other quantum computers try to avoid noise, our system uses the noise,” he says.

As a result, QCI’s EQC does not require any specialised or costly infrastructure, and needs minimal calibration. It is stable and maintains coherence at room temperature, in any environment. So, there is no restriction to where and how it can be deployed.

Xanadu’s Borealis is also a photon based quantum computer, where a quantum light source, with adjustable brightness, emits trains of up to 288 squeezed-state qubits.

Honeywell Quantum Solutions uses trapped ions to achieve their quantum goals. According to their documentation, their systems ‘trap’ charged ytterbium atoms (ions) with electromagnetic fields so they can be manipulated and encoded with information using microwave signals and lasers. Their latest offering, the System Model H1, promises high-quality quantum operations (fidelities) and longer coherence times than other quantum computing technologies.

Honeywell has collaborated with Cambridge Quantum, which makes AI based quantum software and algorithms, to form Quantinuum, to offer an integrated end-to-end quantum platform that is entirely platform agnostic. Their first product, Quantum Origin, is the world’s first cryptographic key generation platform based on verifiable quantum randomness. They are also developing solutions for chemistry, natural language processing, machine learning, and more.

IonQ, the first quantum computing startup to go public, uses individual atoms as their qubits. According to company literature, “At IonQ, we take a different approach, and use a naturally occurring quantum system: individual atoms. These atoms are the heart of our quantum processing units. We trap them in 3D space, and then use lasers to do everything from initial preparation to final readout.”

From here and there

- Intel, together with collaborators, recently created the first silicon qubits at scale at its D1 manufacturing factory in Hillsboro, Oregon. The result is a process that can fabricate more than 10,000 arrays with several silicon-spin qubits on a single wafer with greater than 95% yield. The qubit count and yield are much higher than the typical university and laboratory processes used today.
- Scientists at the Harvard-MIT Center for Ultracold Atoms have demonstrated a programmable quantum simulator capable of operating with 256 qubits. The system uses arrays of highly-focused laser beams, to trap individual qubits and drag them into desirable arrangements to perform meaningful calculations. This will help to build large-scale, practical quantum computers with great applicability, in the near future.

Despite all the advancements, Prof. Dumke feels that, “Hardware is and will be always the bottleneck. This is true for any technology, and for classical computers as well as for quantum computers. The reason why we do not have a universal quantum computer yet is that the hardware is not mature enough. I can foresee that specific hardware solutions made from noisy processors will be tailored to solve special use-cases in the near future.”

Making quantum more accessible

More than pure quantum tech advancements, what is more interesting today are the different approaches being taken to make quantum computing accessible to more people. Two of the most significant efforts in this direction involve offering hybrid solutions, which combine quantum computing with classical computing setups; and full-stack cloud solutions that mask the complexity and let more people tap the benefits of quantum.

Liscouski says, “Our goal from the beginning has been to make quantum computing as available as possible so that business users with no quantum expertise can benefit from it. That’s why we invented Qatalyst, a ready-to-run quantum optimisation and machine learning software that masks the massive complexity and effort needed to solve even basic problems on quantum computers.”

Through QUBT U, they also provide the Qatalyst software to students,

allowing them access to the world of quantum computing without having to be quantum programmers! “We believe that workforce development is a key component of our responsibility to our society,” says Liscouski.

IBM’s plans to build a 4,000+ qubit processor by 2023 goes hand-in-hand with significant milestones to build an intelligent quantum software orchestration platform that will abstract away the noise and complexity of quantum machines, and allow large and complicated problems to be easily broken apart and solved across a network of quantum and classical systems. “There is an inherent strong coupling of quantum and classical computers. Both paradigms of computing—classical and quantum—will work seamlessly together to solve different parts of complex computing problems that are best suited for their respective strengths and capabilities,” says Subramaniam.

IBM is integrating quantum computing with high-performance computing and hybrid cloud technologies in serverless implementations that remove the complexity of infrastructure management, putting the focus on coding only.

“It took 60 years to abstract software in classical computing to the point where users could input a simple line of code into a templated program in order to build an app or website. Quantum computing is going through a similar process—within the decade. IBM

released the Qiskit Optimization Module 2020 as a first step in making frictionless quantum computing a reality by 2025.

“In the future, a program will be handed off to the cloud and vast quantum and classical resources will be employed, and within the blink of an eye the solution will be returned perfectly optimised. The Qiskit Optimization Module enables easy, efficient modelling of optimisation problems for developers and optimisation experts without quantum expertise. It uses classical optimisation best practices and masks complex quantum programming,” says Subramaniam.

IBM has installed on-site quantum systems in Germany and Japan, with plans for installations in Korea, Canada, and in the USA at the Cleveland Clinic. They also have over 20 systems in their Poughkeepsie and Yorktown locations in the USA, which can be accessed over the cloud.

In 2016, IBM became the first company to put quantum computers on the cloud. This was a turning point. “Since then, we have actively built up an active community of more than 400,000 users and 190+ organisations in the IBM Quantum Network that are running more than 4 billion circuits every day and exploring practical applications to realise the wide-ranging benefits of this technology for business and society,” says Subramaniam. From India, IIT Madras became the first institution to join the IBM Quantum Network, in September 2022.

“Companies such as IBM and Microsoft have built many sophisticated simulators and software stacks that are interoperable with different QPUs. There are also startups like D-Wave and Xanadu, which have got their own software platforms. Amazon Web Services (AWS) is attempting to become QPU-agnostic through its Braket portal. Most recently, Nvidia has stepped into this space. So, there are a lot of healthy alternatives,” says Prof. Prabhakar.

Microsoft Azure Quantum is a hardware-agnostic platform, which brings together a range of quantum

computing and optimisation solutions into a single cloud service. You can write code and run it on the quantum hardware of your choice. QCI, IonQ, and Quantinuum are some of those who have partnered with Azure. NASA's Jet Propulsion Lab used Azure Quantum to develop an optimisation solution that helped reduce scheduling times from hours to minutes. Ford and Microsoft teamed up to use quantum methods to figure out how to reduce traffic congestion in Seattle.

Google's Cirq is a Python software library for writing, manipulating, and optimising quantum circuits, and then running them on quantum computers and quantum simulators. Cirq provides useful abstractions for dealing with NISQ computers, where details of the hardware are vital to achieving useful results.

Amazon Braket provides customers access to quantum computing technologies from multiple quantum hardware providers, including D-Wave, IonQ, and Xanadu. It also offers a consistent set of development tools for you to build and test quantum computing projects.

Prof. Prabhakar adds that most of the software is built around an open source effort. "These are still early days, and a collaborative approach is likely to yield higher dividends. However, there is a danger of having too many alternatives for a beginner to comprehend. It helps that most platforms rely on Python libraries, making them amenable to a quick introduction," he says.

Applications galore

Although quantum computing is quite a few years away from becoming mainstream, it is time for companies to start exploring what it can do for them. Liscouski lists out a few key industrial applications, for us to understand the potential of quantum computing to transform businesses:

Supply chain and retail:

- Quantum computing enables users to improve multi-level

With simulators available, who needs quantum computers?

In 2019, Google researchers claimed they had achieved quantum supremacy when their quantum computer Sycamore performed an abstruse calculation, which would take a supercomputer 10,000 years to solve, in 200 seconds. Recently, a group of scientists in China created waves by completing the same computation in a few hours with ordinary processors. They recast the problem as a 3D mathematical array called a tensor network, and ran the simulation by multiplying all the tensors.

Quantum simulators are software programs that allow you to use a classical computer to run quantum circuits as if they were being run on a quantum computer. There are several simulators like Intel-QS and NVIDIA cuQuantum. And the question often arises as to why we need quantum computers when these simulators are there. While simulators can probably solve some problems, solving actual quantum problems using them would require an unmanageably high amount of computing resources. That said, simulators can be helpful in developing quantum computing applications and algorithms, as they can help debug code and test near-term systems.

replenishment throughout the entire supply chain from the manufacturer to the distribution centres, right down to local stores.

- Users can accelerate warehouse performance with highly optimised pick order and path routing, while accommodating the use of more autonomous robots (as well as the growing data complexity that comes with them).
- Quantum computing can perform inter-modal optimisations (for example, across long-haul and local surface freight) that would be very challenging for classical computers.

Cybersecurity:

- Cybersecurity is a perfect application for quantum computing because of the complexity of unstructured data that must be repetitively and efficiently analysed to prevent threats and respond quickly to intrusions.
- Quantum computing enables the real-time investigation of network intrusion to disrupt and prevent data exfiltration incidents.
- It also has the ability to detect patterns of divergent network traffic.

Quantum sensing:

- Quantum light detection and ranging (lidar) will extend the capabilities of

existing lidar to provide enhanced data acquisition and analysis.

- Quantum sensing is also applicable in the medical imaging field; imagine being able to see below the skin without high power such as X-ray.

Life sciences and pharmaceutical:

- Quantum computers can assist in a wide range of application areas where understanding interrelatedness of common attributes is key to discovery. For example, identifying patient attributes and subpopulations in clinical trials.
- Quantum computing can help discover heterogeneous networks of proteins and genes implicated in human biological function and disease states or in viral or microbial life cycles.
- It can also analyse networks to compare molecular substructures and analyse the larger molecules necessary in pharmaceutical research.

"The early adopters are financial institutions seeking an advantage in wealth management, logistics companies attempting to reduce costs and improve efficiencies, and pharma companies that need to tackle the quantum chemistry of their new molecules. We also expect telecom and data security companies to be watchful of any breakthroughs in cryptography,

Stay Connected. Stay Informed. Stay Ahead.



SUBSCRIBE AND SAVE

ORDER FORM

PRINT MAGAZINE	1 YEAR (12 copies each)	3 YEARS (36 copies each)	5 YEARS (60 copies each)
Electronics For You (Rs 100/copy)	WITHIN INDIA (IN RUPEES)		
	840 <input type="checkbox"/>	2150 <input type="checkbox"/>	3000 <input type="checkbox"/>
	SAARC COUNTRIES (IN US\$ BY AIR)		
	50 <input type="checkbox"/>	135 <input type="checkbox"/>	NA
OTHER COUNTRIES (IN US\$ BY AIR)			
100 <input type="checkbox"/>	270 <input type="checkbox"/>	NA	

PRINT MAGAZINE SUBSCRIBERS GET:

- Free e-magazine every month
- Free delivery of print magazine by post
- And much more (check: subscribe@efy.in)
- For delivery by courier, please add Rs 50 for each copy

To subscribe online, visit
<https://payment.efyindia.com>

OR
SCAN
THIS
CODE



e-magazine subscriptions within India are available at half the rates mentioned here.
Overseas rates for each e-magazine in US\$: 1 year: \$12; 3 years: \$33; 5 years: \$50 only

Name _____ Organisation _____ Mailing Address _____

City _____

Pin Code _____ State _____ Phone No. _____ Email _____

Subscription No. (for existing subscribers only) _____ I would like to subscribe to the above (✓)marked magazine(s) starting with the next issue. Please find enclosed a sum of Rs _____ by DD/MO/crossed cheque bearing the No. _____ dt. _____ in favour of EFY Enterprises Pvt Ltd, payable at Delhi.

Please mark one (nearest) relating to your subscription: Indian Company MNC R&D organisation Engineering institute College/School Any other (specify): _____

Send this filled-in form or copy to : EFY Enterprises Pvt Ltd, D-87/1 Okhla Industrial Area, Phase 1, New Delhi 110 020 | Ph: 011-40596600 | e-mail: support@efy.in

Terms:- # These rates are applicable for new subscribers as well as renewal by existing subscribers # Please allow 4-6 weeks for processing of your subscription.
Please include your pincode for prompt delivery of your copy.

Experts



Prof. Dr. Anil Prabhakar,
Department of Electrical
Engineering, IIT Madras



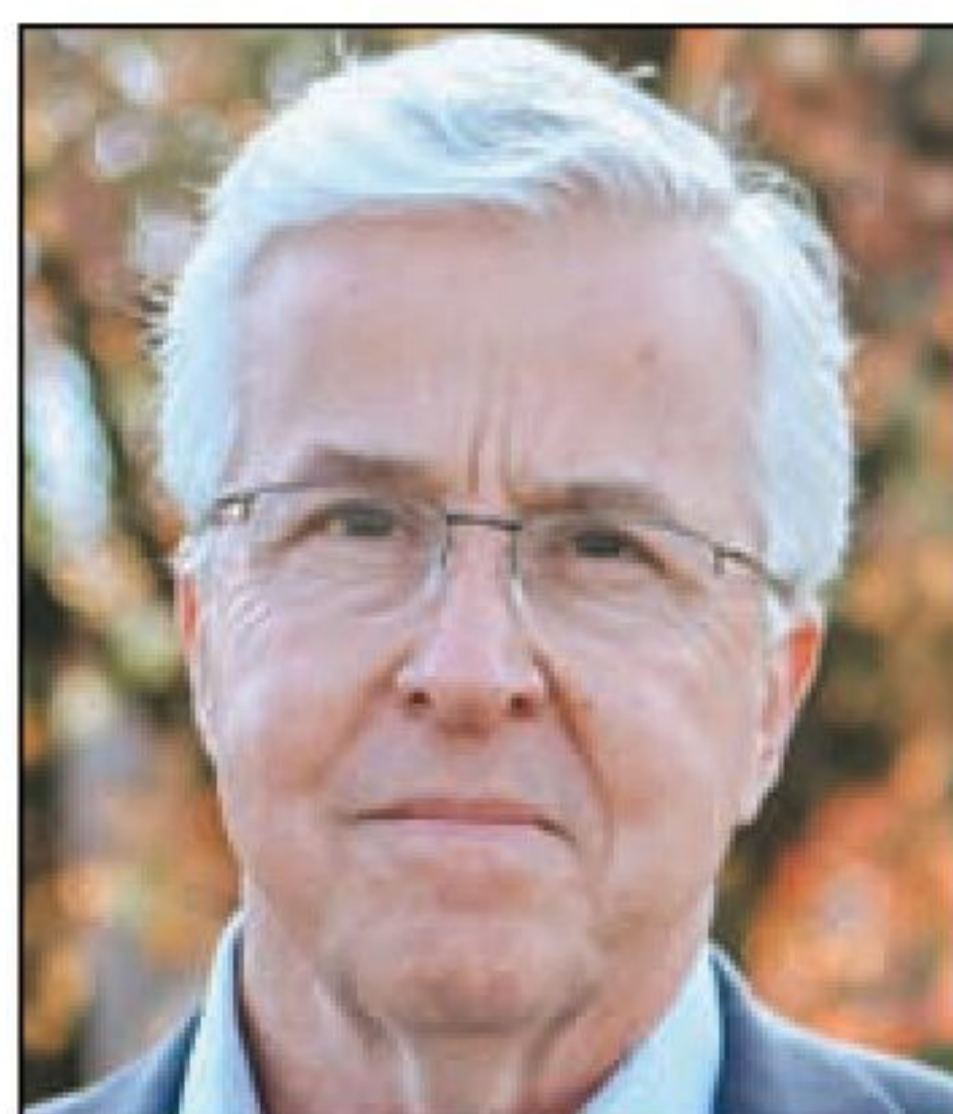
L. Venkata Subramaniam, Senior
Manager - AI Science and IBM
Quantum Ambassador, IBM
Research India



Rahul Mahajan,
CTO, Nagarro



Prof. Rainer Dumke, Principal Investigator at the
Centre for Quantum Technologies, Professor at
Nanyang Technological University, Singapore,
and Lead for hardware at Singapore's National
Quantum Computing Hub



Robert Liscouski, CEO,
Quantum Computing Inc.
(QCI)

providers, 15-20 startups and 40-50 academic institutions active in this domain. It says that the industry would add about \$310 billion to the Indian economy by 2030 and sectors such as manufacturing, high-tech, banking, and defence will likely lead the charge of adopting quantum technologies for critical and large-scale use cases.

Achyuta Ghosh, Research Head at NASSCOM, said in an *Economic Times* interview, that companies like Tata Consultancy Services, HCL Technologies, Infosys, Tech Mahindra, Zensar, Mphasis, and Coforge were creating use cases for quantum technologies and proof-of-concept for clients. There are also some interesting startups in this space such as QNu Labs, Taqbit Labs, QRDLab, and QpiAI Tech.

There are around a 100 quantum projects initiated in India, of which 92% are funded by the government. In August this year, the Ministry of Defence announced that the Indian Army had started the process of procurement of Quantum Key Distribution (QKD) technology developed by a Bengaluru based cybersecurity company by issuing a commercial request for proposal.

In a global survey of 750 companies conducted by KPMG early this year, a quarter of the business decision-makers said they already had quantum computing projects in place, one-third had either an internal team or external advisers looking into how they can use the technology, and 31% of organisations said they were discussing how they will leverage the technology in the future. KPMG reported that only 9% of the respondents said they were not thinking at all about how they can take advantage of quantum. We will leave you thinking about that! **END!** 

and hence invest in quantum communications and post quantum cryptography,” says Prof. Prabhakar.

“IIT Madras has been an early mover in quantum science and technology. We had India’s first quantum key distribution demonstration as early as 2008. Today, we have about 20 faculty and their students and staff looking at different aspects of quantum science and technology. This has developed into a healthy interdisciplinary ecosystem, and is formally known as the Centre for Quantum Information, Communication and Computing. We continue to drive both basic and translational research in collaboration with start-ups and our partners in an Industry-Academia Consortium model, allowing members to leverage common knowledge while pushing specific applications of interest to them,” he adds.

A good quantum of hope

“Hardware with 100+ logical qubits is now available. Hardware with a few 1000s of qubits is not that far, and we

may see such machines in the next 2-3 years. On the software part, we will continue to see improvements in the quantum programming platforms,” says Mahajan.

“With the current noisy systems, we are only able to solve some artificial toy models without practical relevance. However, I am sure that in the next generation of systems we will be able to really realise quantum processors that make an impact and can outperform their classical counterparts on real world problems,” remarks Prof. Dumke.

According to a recent NASSCOM report, India too is gearing up for the quantum era, with about 10-15 government agencies, 20-30 service

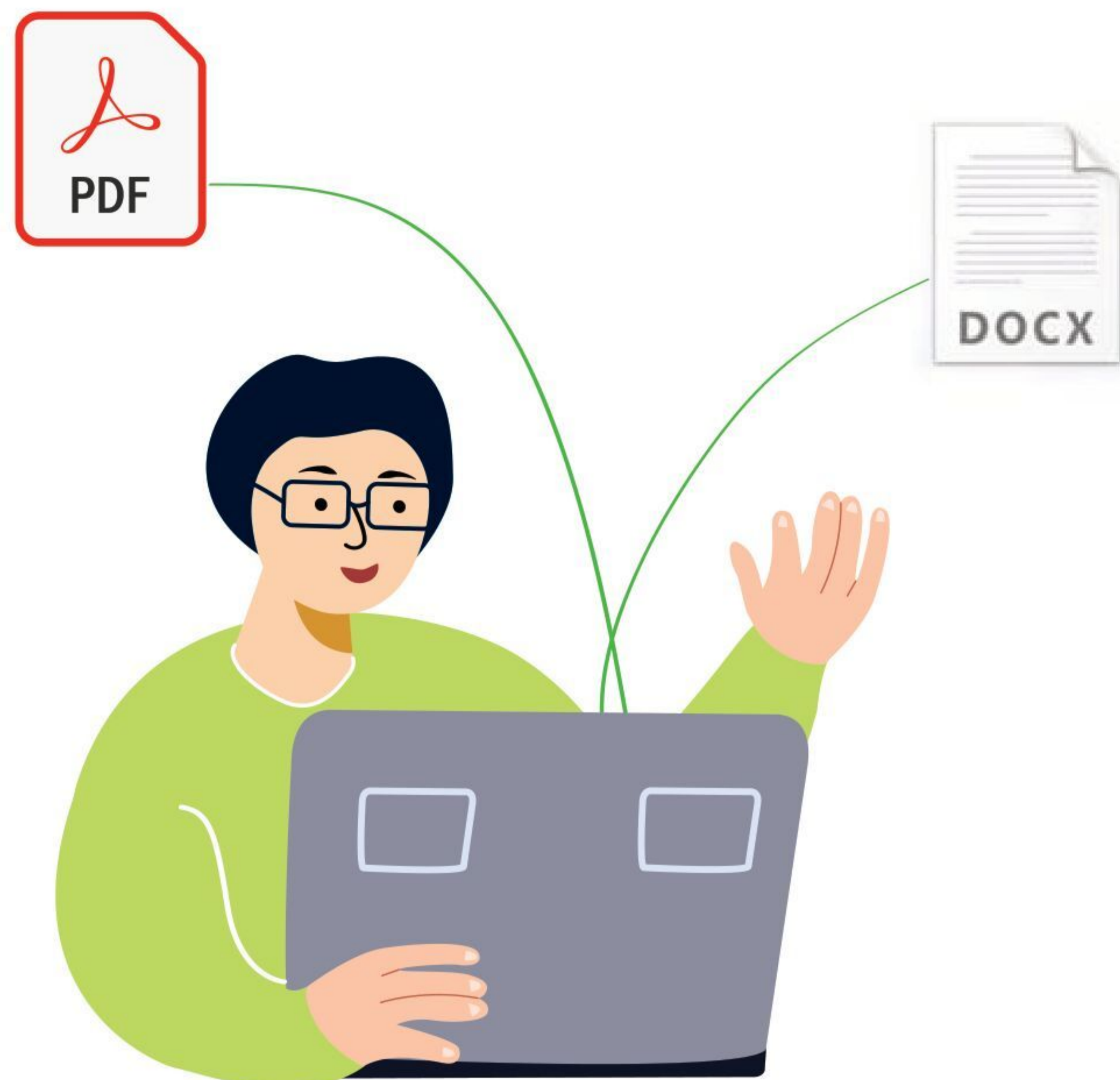
 By: Janani G. Vikram

The author is a freelance writer based in Chennai, who loves to write on emerging technologies and Indian culture. She believes in relishing every moment of life, as happy memories are the best savings for the future.

The article was originally published in the November 2022 issue of Electronics For You.

How to Create PDFs and DOCX Files for Free

Free and open source software can help you automate document creation in the most popular formats.



Governments, businesses and other organisations create a lot of documents. When they automate their systems, they purchase software components that cost hundreds or even thousands of dollars per year or per computer. It seems like a criminal waste.

I have self-published over two dozen books. They were all created using only free and open source software (FOSS). I find that other self-published authors have spent thousands of dollars to publish just one book. This does not include marketing. I am horrified when I read or hear about their expenses.

Why not pay some attention to free choices before splurging so much money on proprietary software? In this article, I will describe how anyone can create PDF and DOCX files for free. The entire toolchain will be FOSS. This process can be easily automated and integrated with application servers.

Use CommonMark (Markdown) for the source document

You can write your source documents in plain text, markup, markdown or directly in rich text. HTML (HyperText Markup Language) is markup. It is displayed as rich text. Rich text is what you see in a browser or document editor such as LibreOffice Writer. Markup is what you see when you do *View » Source* on a web page. The ODT file that you create in Writer is actually a renamed zip file with mostly XML files (eXtended Markup Language).

Markdown is the opposite of markup. It uses ordinary plain-text enhancements to unobtrusively mark up text. Before the World Wide Web came into existence, the internet was mostly email. Email, BBS and Usenet (newsgroups) users developed a plain-text style of formatting for their messages. For example, bold type was wrapped in ***asterisks*** and italicised text was wrapped in *_underscores_*. When John Gruber and Aaron Swartz created Markdown in 2004, they extended this style even further. It is now the most popular form of markdown. Markdown was released as a Perl script. Markdown documents were usually saved with the extension *.md*. I wrote my first book in Markdown and converted it like this:

```
perl markdown.pl jokebook.md > jokebook.html
```

Other implementations of Markdown were based on the Perl script but there were some differences. Finally, in 2019, Jeff Atwood and John MacFarlane published a standardised implementation of Markdown called CommonMark. I learned about it sometime in 2020 and earned the bragging rights for the first book on CommonMark. It is called ‘CommonMark Ready Reference’. Its digital version is available for free on almost all ebook stores (except Kindle). However, I will give you a quick intro so that you have an idea what it looks like.

Chapter Title

Subheading

Bolded text.

Sub-subheading

Italicized text.

1. Ordered list item 1
2. Ordered list item 2
3. Ordered list item 3
4. Ordered list item 4

Sub-sub-subheading

Bold and italicized text.

- * Unordered list item 1
- * Unordered list item 2
- * Unordered list item 3
- * Unordered list item 4

![Image description](image-filename.png)

```
> Blockquoted text or a text quotation
>
> Usually indented on the left.
```

[www.VSubhash.in](http://www.vsubhash.in/commonmark.html "About my book CommonMark Ready Reference")

Figure 1: Markdown uses plain-text enhancements to mark up text

Figure 1 shows how Markdown uses plain text enhancements to mark-up text. A more comprehensive quick-reference card for CommonMark/Markdown is available for free from my website as a printable PDF.

CommonMark has a proper specification. It also has an implementation written in C that is blisteringly fast. I have compiled the source, and offer the Linux and Windows executables from my website. You can find the build instructions in the ebook.

After compilation or downloading the executable, you can convert your Markdown/CommonMark source documents like this:

```
commonmark --unsafe --validate-utf8 jokebook.md > jokebook.html
```

CommonMark can generate headings, paragraphs, blockquotes, images, links, lists, code spans and blocks, horizontal breaks and line breaks. That is about it. It cannot generate tables and other fancy stuff. If you want those, you can write raw HTML and use the *-unsafe* option. By default, CommonMark omits raw HTML to protect software systems from code injections.

The HTML that the CommonMark executable or the Markdown Perl script generates is validation-safe and well-structured. However, it will not have HTML, HEAD, TITLE or BODY tags. The executable's sole purpose is to create HTML markup that can be straightaway used in a pre-existing page or HTML template.

Imagine that this is the markdown source document:

```
Science Jokes
```

```
* **How many astronauts would it take to a screw a lightbulb?**
  One to turn the bulb and several to prevent the spacecraft from spinning in the same direction.
* **What did one radio wave say to another?**
  "You are interfering with my work."
* **What's a radio engineer's favourite food?**
  A can of tuna....
```

CommonMark can be used to convert it like this:

```
echo '<!DOCTYPE html><html><title>2020 Jokebook</title></head><body>' > jokebook.html
commonmark --unsafe --validate-utf8 jokebook.md >> jokebook.html
echo '</body></html>' >> jokebook.html
```

The output HTML will look like this:

```
<!DOCTYPE html><html><title>2020 Jokebook</title></head><body>
<h2>Science Jokes</h2>
<ul>
<li><strong>How many astronauts would it take to a screw a lightbulb?</strong><br />
One to turn the bulb and several to prevent the spacecraft from spinning in the same direction.</li>
<li><strong>What did one radio wave say to another?</strong><br />
" You are interfering with my work."</li>
<li><strong>What's a radio engineer's favourite food?</strong></li>
```

```
strong><br />
A can of tuna.</li>
</ul>
</body></html>
```

CommonMark-generated markup starts from `<h2>` and ends in ``. The rest is the HTML template. Figure 2 shows what the HTML looks like in a browser.

Use LibreOffice to create ODT, DOCX and PDFs

You already know that LibreOffice is the FOSS alternative to Microsoft Office. It has a word processor Writer (the Word alternative), spreadsheet application Calc (the Excel alternative), presentation slide maker Impress (the PowerPoint alternative), and a few other applications. While LibreOffice hoots and toots like a regular GUI application, it also has a demure command-line interface.

To convert the afore-mentioned HTML document to ODT format, type:

```
libreoffice --convert-to "odt" jokebook.html
```

You can use the same HTML document and convert it to DOCX so that Microsoft Office users can feel happy. (Microsoft Word can edit ODT files just fine.)

```
libreoffice --convert-to "docx:MS Word 2007 XML" jokebook.html
```

You can use the ODT or DOCX file that you generated and convert it to a PDF file:

```
libreoffice --convert-to "pdf" jokebook.odt
```

Why not convert to PDF straight from the HTML? Why create the intermediate ODT or DOCX file? Because the HTML document does not have any concept of page size, margins, headers, footers and all that jazz.

Create documents with images

When you convert a HTML document containing images, the resultant ODT or DOCX documents will show the images all right. When you move the documents or mail them to someone, the images will disappear. This is because the images in the ODT or DOCX documents continue to be loaded from the source image files. To fix this problem, you need to encode the images as text. This is similar to how images and attachments are encoded in email messages — using base64 encoding. View the message source of an email containing an attachment; you will find the file encoded as plain text.

Instead of using an image file like this:

```

```

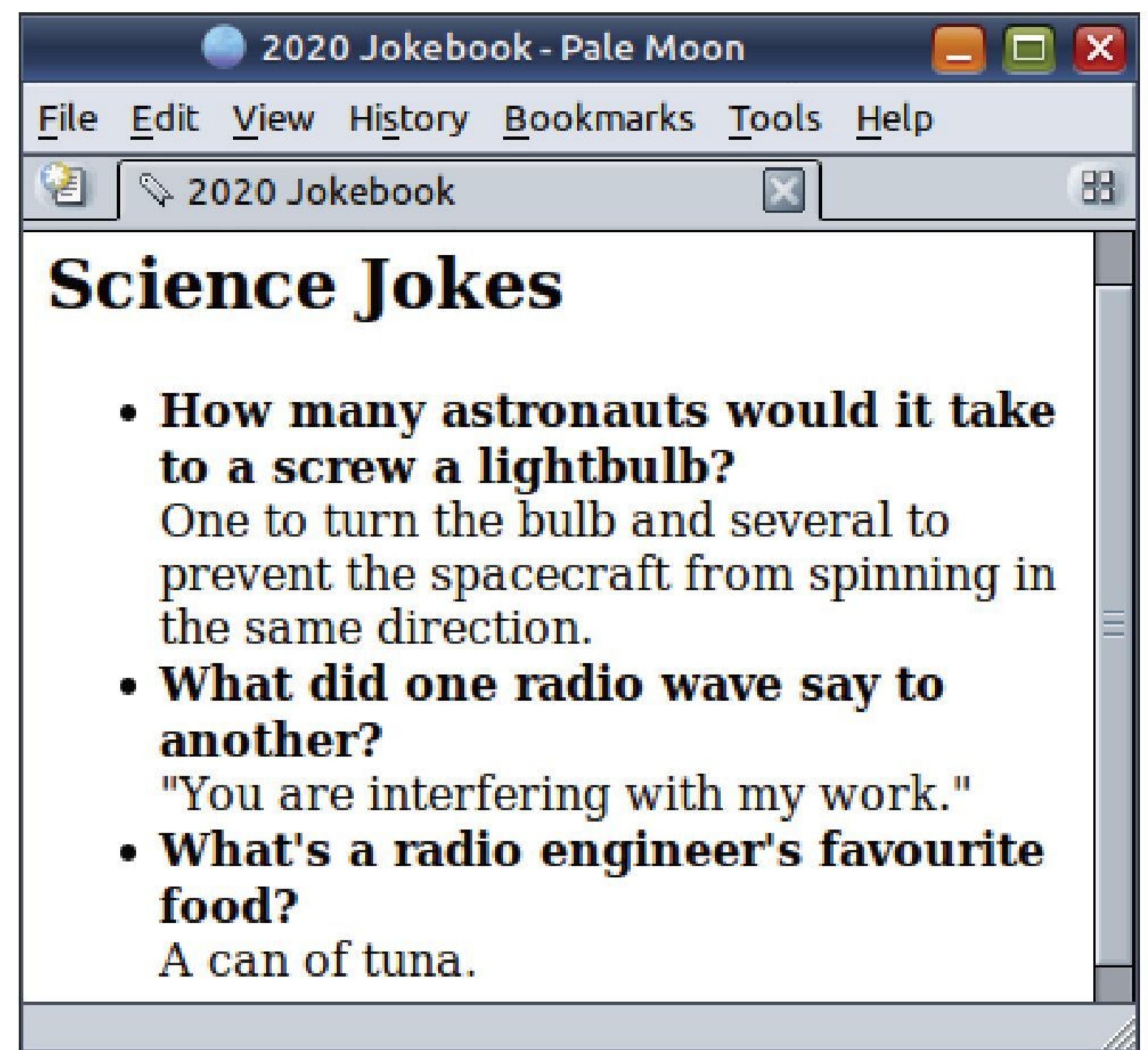


Figure 2: CommonMark generates HTML tags on a strictly as-needed basis. You have to use that generated content in a pre-existing page or HTML template

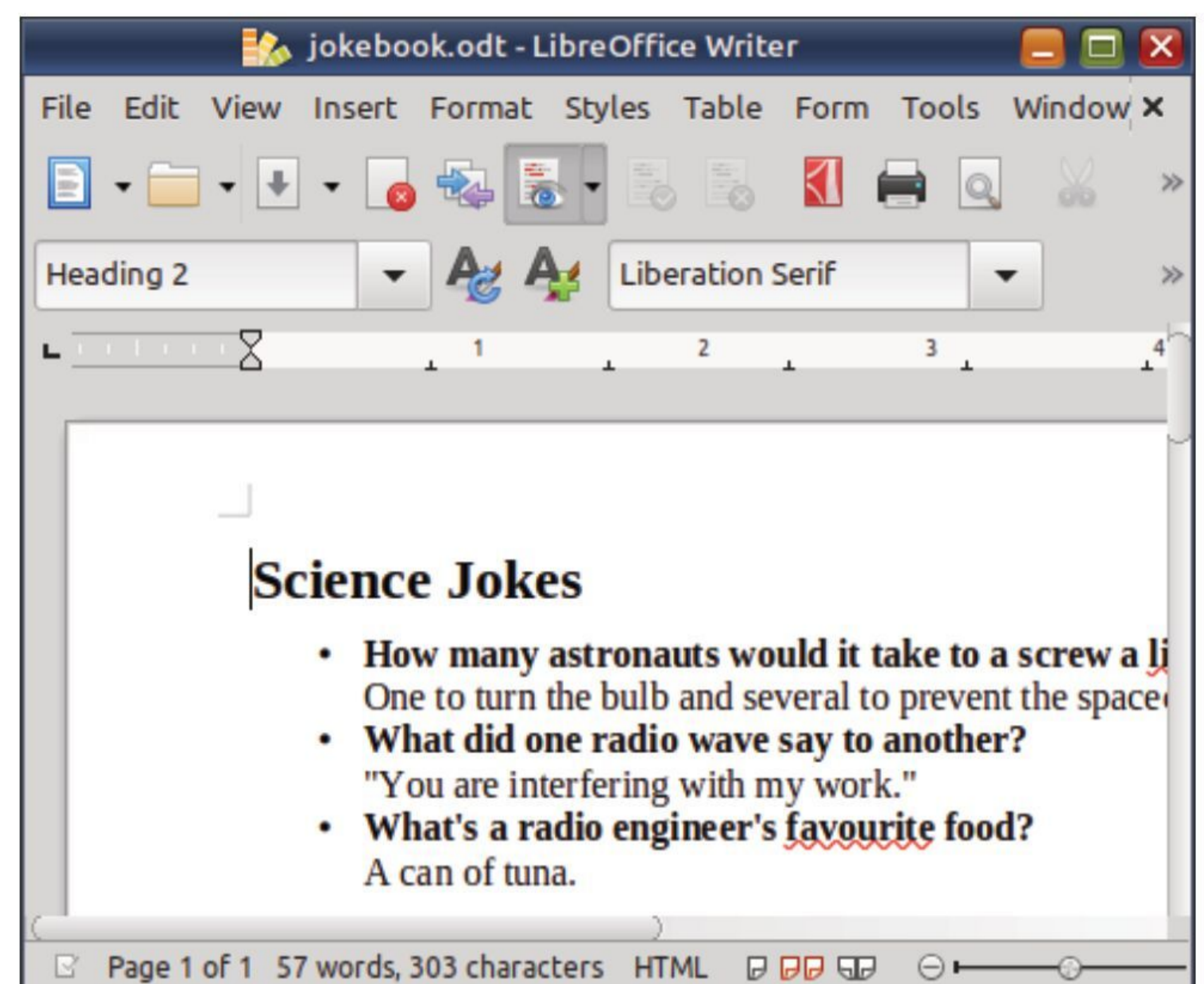


Figure 3: The command-line interface of LibreOffice can be used to automate the conversion of HTML documents to ODT, DOCX or PDF documents

... you can encode it as text like this ...

```

```

That is not all. I have truncated the actual text of the encoded image. The full text is nearly 600 lines. If you are curious, try a command like this:

```
base64 lion-and-deer.png
```

You do not have to dirty your hands with manual text-encoding. LibreOffice will encode the images as text.



Asia's #1 Open Source Conference

Total registrations

10,340

Total attendees

3732

Speakers

90+



19th Edition

OPEN

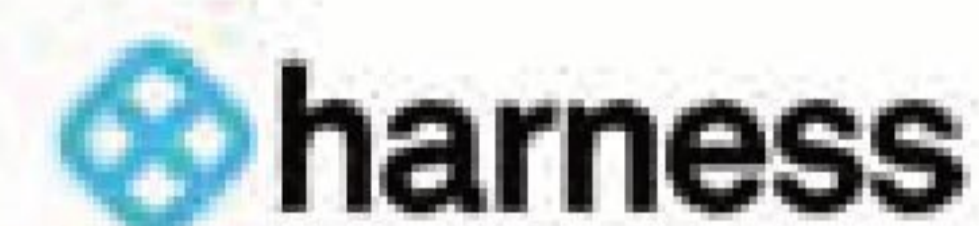
SOURCE INDIA

Thank You

VISITORS, for making the event a huge success

SPEAKERS, for contributing your valuable thoughts

PARTNERS, for your support



For more details, call on +91-98111-55335 or email us at info@opensourceindia.in

```
echo '<!DOCTYPE html><html><title>2020 Jokebook</title></head><body>' > jokebook.htm
commonmark --unsafe --validate-utf8 jokebook.md >> jokebook.htm
echo '</body></html>' >> jokebook.htm
libreoffice --convert-to "html:HTML:EmbedImages" jokebook.htm
```

Here, the *.htm* file (referring to an external image) was created by CommonMark. LibreOffice consumed that *.htm* file and created a *.html* file with the text-encoded image. This self-contained HTML is now portable and not dependent on any external files. When you convert such an HTML document, the resultant ODT or DOCX file will also be self-contained and portable. Even if you delete the image file, you will still be able to see it in the *.html*, *.odt* or *.docx* files.

```
libreoffice --convert-to "html:HTML:EmbedImages" jokebook.htm
libreoffice --convert-to "odt" jokebook.html
libreoffice --convert-to "docx:MS Word 2007 XML" jokebook.odt
libreoffice --convert-to "pdf" jokebook.odt
```

Enhanced document content

As mentioned earlier, CommonMark outputs only a limited set of HTML tags. For creating content that it does not support, you will have to add raw HTML in your markdown.

Animal Jokes

```
* **Why did the lion cross the road?*
```

```
Because <span style="color: white; background-color: red; border-radius: 0.5em; border: 2px dashed yellow; ">the buck stops here</span>.
```

```
![Lion and deer](lion-and-deer.png)
```

Do not go overboard with this raw HTML content. LibreOffice has its own limited set of HTML tags and CSS styles that it can convert.

What does the screenshot in Figure 6 say? LibreOffice does not do rounded corners, among other things. So, temper your excitement.

This use of raw HTML is crude. It defeats the idea of markdown. The purpose of CommonMark is to create a well-structured document. Special styling can be affected by including CSS styles in the HTML template. I leave that as homework for you or your developers. (Just brush up on CSS pseudo-classes, selectors and attribute matching.) I gave the above example to make the concept easy to understand.

Unlike styles, tables are not fancy stuff. They are the nuts and bolts of financial documents. For those, you can use raw HTML inline. LibreOffice will convert HTML tables all right.

What about headers and footers? Does CommonMark support them? Can they be added to the HTML template similar to CSS



Figure 4: A text-encoded image is a huge chunk of unintelligible but plain text

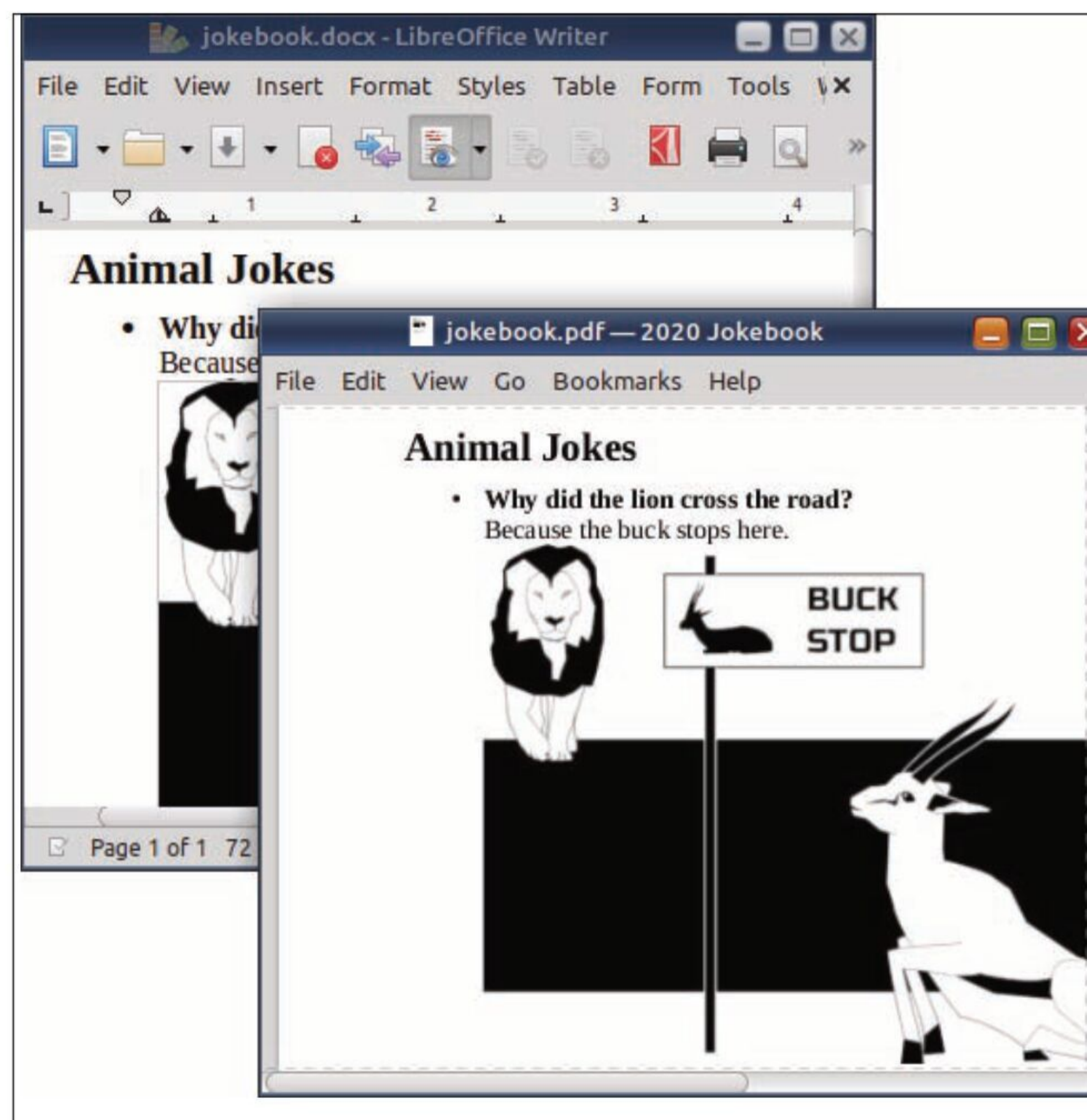


Figure 5: To make ODT and DOCX files self-contained or portable, images in the source HTML files should be text-encoded. PDF files use self-contained images, which makes them portable by definition

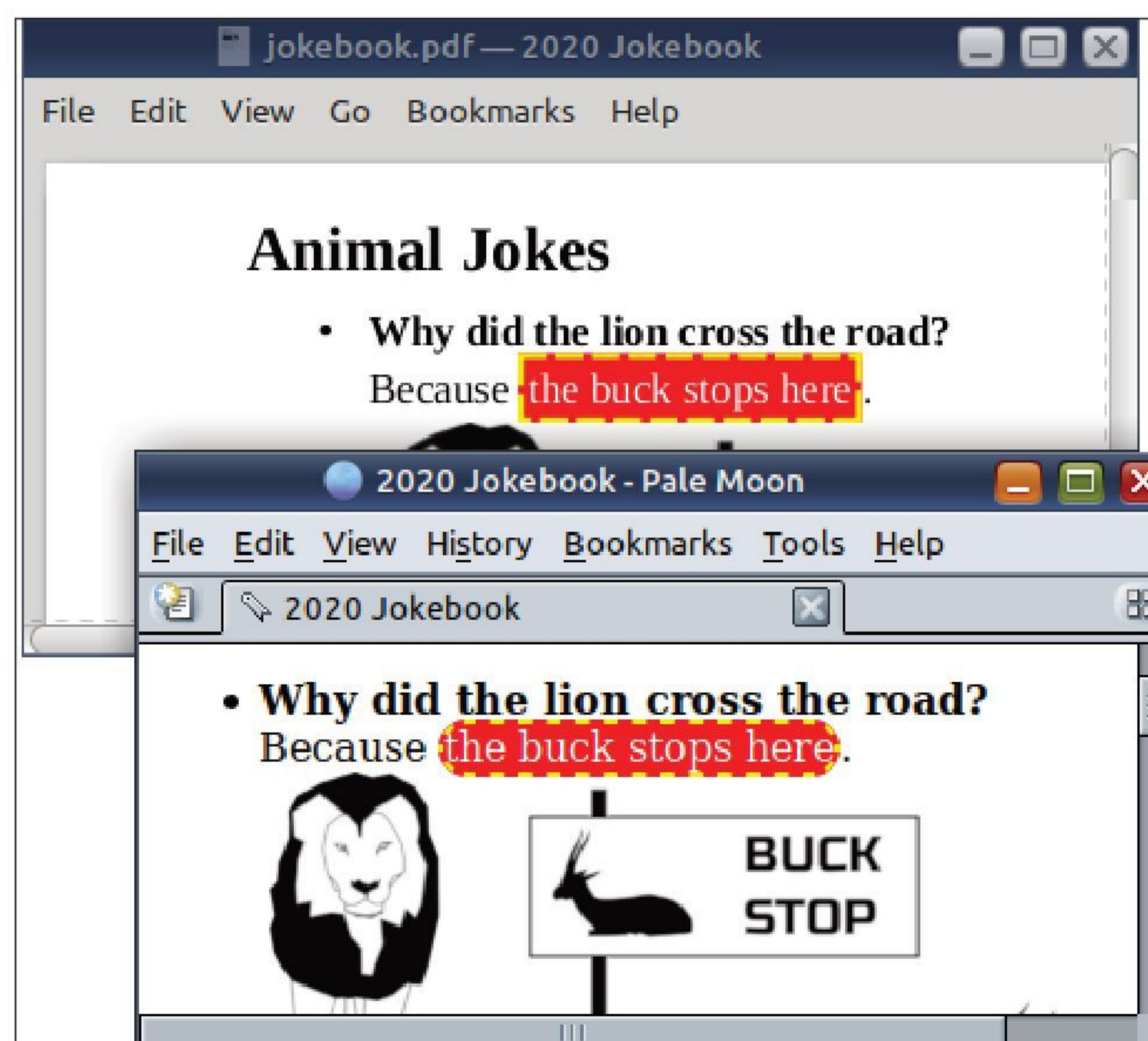


Figure 6: LibreOffice supports a limited set of HTML tags and CSS styles that it will convert to ODT, DOCX or PDF. For more sophisticated documents, use wkhtmltopdf

styles? Unfortunately, no. This kind of toolchain is good for one-page documents. Or, multi-page documents without headers and footers. If you want those, then you have to use a different tool called wkhtmltopdf. This is essentially a headless Firefox browser that can convert heavily formatted HTML documents to PDF. You can specify headers and footers using separate HTML files. I created all 29 of my books using this tool. It supports lots of CSS styles that LibreOffice does not support. My books would not look so rich if I relied on LibreOffice. (wkhtmltopdf is based on an old Firefox codebase that is not being updated. A lone Indian is maintaining it. It has a few bugs. Read my Unicode tricks article on how I fixed them.) Also, unlike LibreOffice, wkhtmltopdf can execute JavaScript. It has an option for adding a few seconds of delay so that the JavaScript can do its thing before wkhtmltopdf starts printing the document to PDF. So, for simple documents, use LibreOffice. And, for heavy-duty documents, use wkhtmltopdf. Both these programs will use the headings in the source document to create the bookmark tree in the PDF document. LibreOffice has one advantage that wkhtmltopdf does not have — it can create ePUB ebooks and several other types of documents.

Additional document features

If you have to merge two ODT or DOCX documents, I do not know of or care for any tool to do it. Do the merging in the markdown source material and then create the combined document.

For PDFs, there are lots of tools. For my books, I have to combine some page-size images with the PDF of the interior pages. For that, I use ImageMagick and pdftk.

```
magick title-page.png -resize 100% front.pdf
pdftk front.pdf jokes.pdf output book.pdf
```

pdftk is a very powerful tool and does more than merging PDF pages or collating them from different documents. It can also watermark and encrypt your PDF.

```
pdftk book.pdf output book-encrypted.pdf \
  encrypt_128bit \
  owner_pw RcHrDsTlMn^012 \
  user_pw FrSfTWrfnDtn^321
```

And, for a touch of class, add metadata to the PDF:

```
echo "InfoBegin" > meta.txt
echo "InfoKey: Title" >> meta.txt
echo "InfoValue: 2020 Jokebook by V. Subhash" >> meta.txt
```

```
echo "InfoBegin" >> meta.txt
echo "InfoKey: Subject" >> meta.txt
echo "InfoValue: Fresh Clean Jokes" >> meta.txt
```

```
echo "InfoBegin" >> meta.txt
echo "InfoKey: Author" >> meta.txt
echo "InfoValue: V. Subhash (&#169; 2022 V. Subhash. All
rights reserved.)" >> meta.txt
```


```
pdftk "jokebook.pdf" update_info meta.txt output
2020-jokebook.pdf
```

Whatever I cannot do with these free PDF tools, I go ahead and write my own custom utility using iText Java. For example, when I found that pdftk was unable to handle some of my bigger books, I used a JAR file executable that I had created using iText.

Use in online applications

Document creation requires a lot of heavy lifting in terms of CPU usage, disk access and memory requirements. Do not integrate these tools directly with web applications. Use a daemon (background process or service) that is launched by the OS on startup to do the document creation. Your web applications should just queue document-creation jobs to this daemon. When the documents are ready, the daemon should invoke an API routine in the web application to notify that the job has been completed. Otherwise, your online users will crash your system when their numbers build up.

Donate

If free software such as those mentioned in this article helped you to reduce costs, then make it a point to donate some money to their projects. If you are an independent software vendor (ISV), then tell your client that you did not write all the code and some part of the system relies on free tools. Tell the client to make donations commensurate to the usage of the free software in that system. The creator of iText has an incredible story (check the *Reference* given at the end of this article) of how Google refused to offer nothing more than a T-shirt and a mug for using his PDF library in their products such as Google Analytics, Google Docs and Google Calendar. Do not be like that. Indian corporations have a statutory obligation to divert a certain amount of their profits to charitable causes. Most of them will be glad to route donations to open source projects as CSR (corporate social responsibility) grants. **END** 

Reference

- <https://entrepreneurd.lowagie.com/chapters/c13.html>

By: V. Subhash

The author has written over two dozen books including Quick Start Guide To FFmpeg (from Apress), Linux Command-Line Tips & Tricks, CommonMark Ready Reference, Cool Electronic Projects, How To Install Solar and Unlikely Stories.

AI Based Tools that Aid Research

AI has been around for a very long time, but the current wave seems to be the biggest and most distinctive because it has brought with it tools that are readily usable. Two of the most complex tasks for a research scholar are to search through volumes of literature and understand the complex contents of research papers. This article presents two tools that make both these tasks easier.



Academic research is a challenging task and requires considerable focus and the ability to handle high levels of cognitive load. There are various phases in academic research that can benefit by using specific tools. The application of proper tools can reduce the academic load manifold and help increase the quality of output.

There are many software available that can assist a scholar in the different phases of research. This article focuses on artificial intelligence or AI based tools. An important part of academic research is formulating the research question. Once you find the right question, things become much clearer. To ask the right question the steps researchers follow are:

- Search existing literature
- Understand the papers
- Find the research gaps
- Construct the question

Searching through literature is not as simple as searching a random query in a search engine. It requires specific skills. Two tools that can help a scholar in surveying the literature effectively and understanding its contents easily are:

- Elicit
- Explainpaper

Do remember, however, that these



Figure 1: AI for scholars -- Elicit and Explainpaper

tools simply assist scholars, and do not automate the research.

Elicit

Elicit is an AI based research assistant, which is based on the GPT-3 (generative pre-trained transformer, third generation) language model. It can assist scholars in searching through the literature and asking the right questions. It has the ability to find answers from 175 million papers. You can run Elicit on your own papers as well as by uploading them.

Elicit is the product of Ought, a non-profit research lab. It currently focuses on machine learning, and is funded by various organisations including Open Philanthropy and Future of Life Institute.

Elicit is capable of assisting the research scholar in the following tasks:

- Searching through the literature to find relevant papers
- Generating initial leads for the paper
- Answering specific questions with the knowledge of papers

Though there are some traditional literature search tools, Elicit is different from them in many ways such as:

- Locating the papers that didn't match with the exact keywords. It can find the relevant papers based on semantic search.
- Elicit has the capability to generate summaries of the abstract. Reading and understanding the abstract of all the papers returned in a search result is a time-consuming, difficult and demanding task. Elicit helps here by providing an automatically generated summary.
- Elicit enables the scholar to navigate through the citation graph in both forward and backward directions. This means that you can easily navigate to the papers that are referenced (backward) in the current paper and the citation (forward) of the current paper.
- An interesting feature of Elicit

is the ability to customise the fields in the search result. You can also add columns of your choice, and sort the information in them.

- Advanced filtering options are provided based on study type. If you are interested in finding research papers that belong to a particular study type such as randomized control trials, systematic reviews, or meta-analyses, you can simply do that by applying such a filter.
- Reference managers such



Figure 2: Elicit literature review workflow features

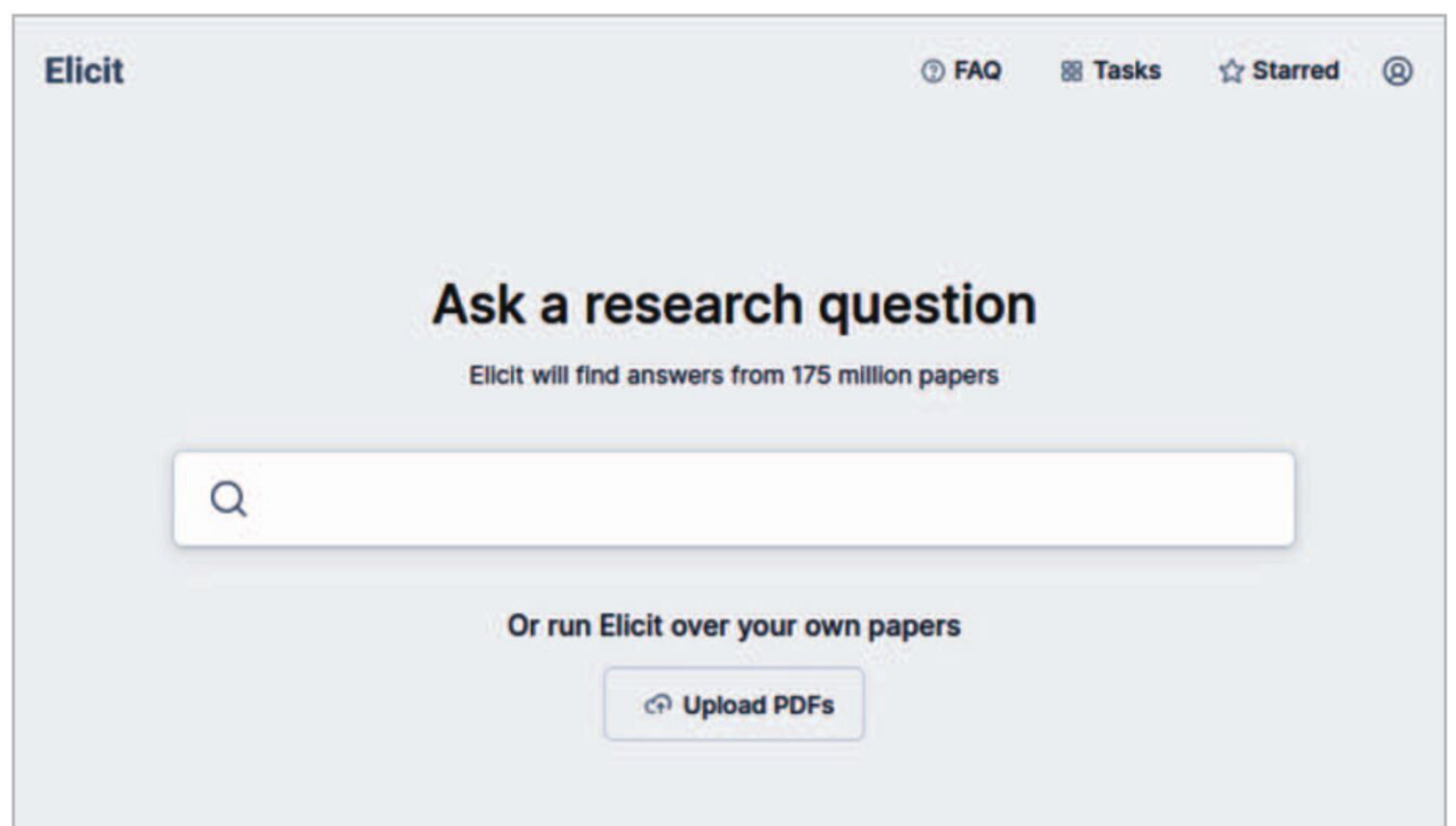


Figure 3: Elicit home page

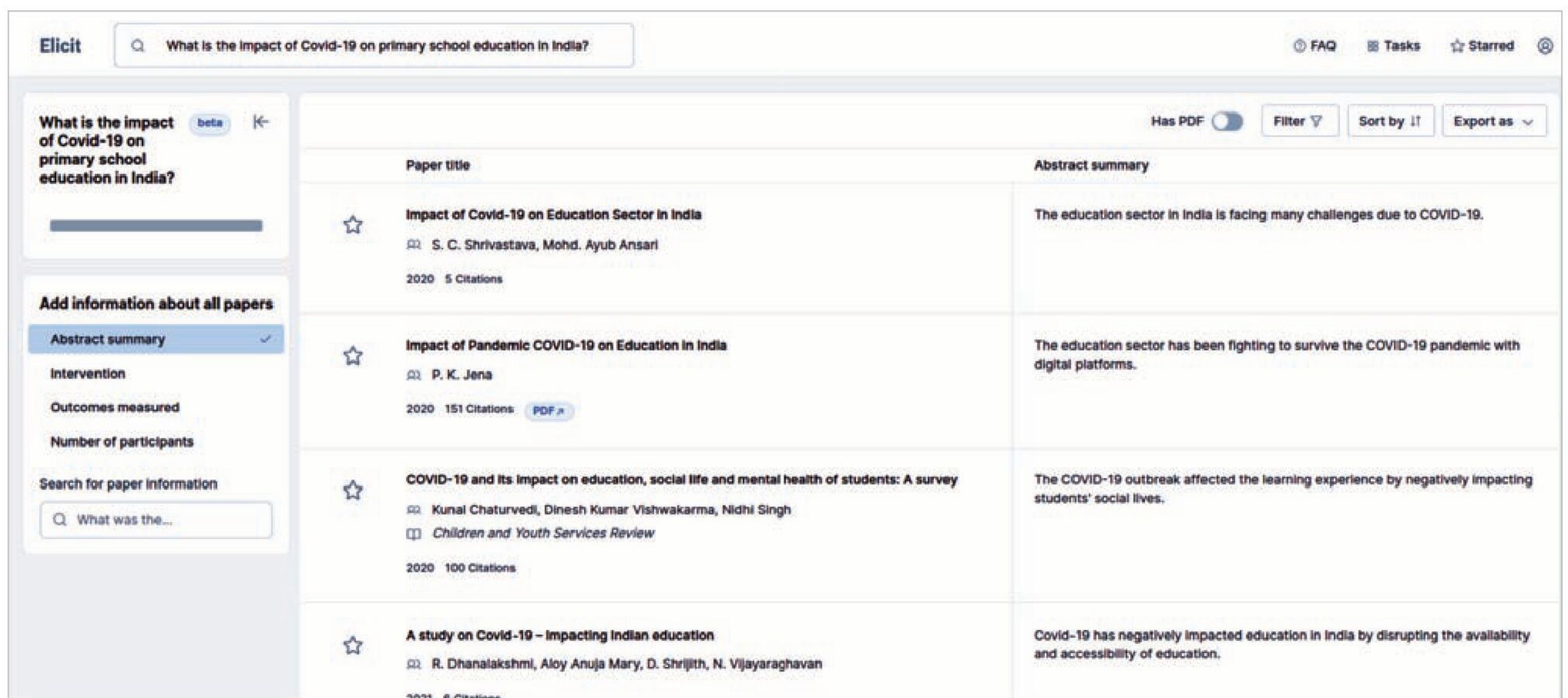


Figure 4: Elicit result page

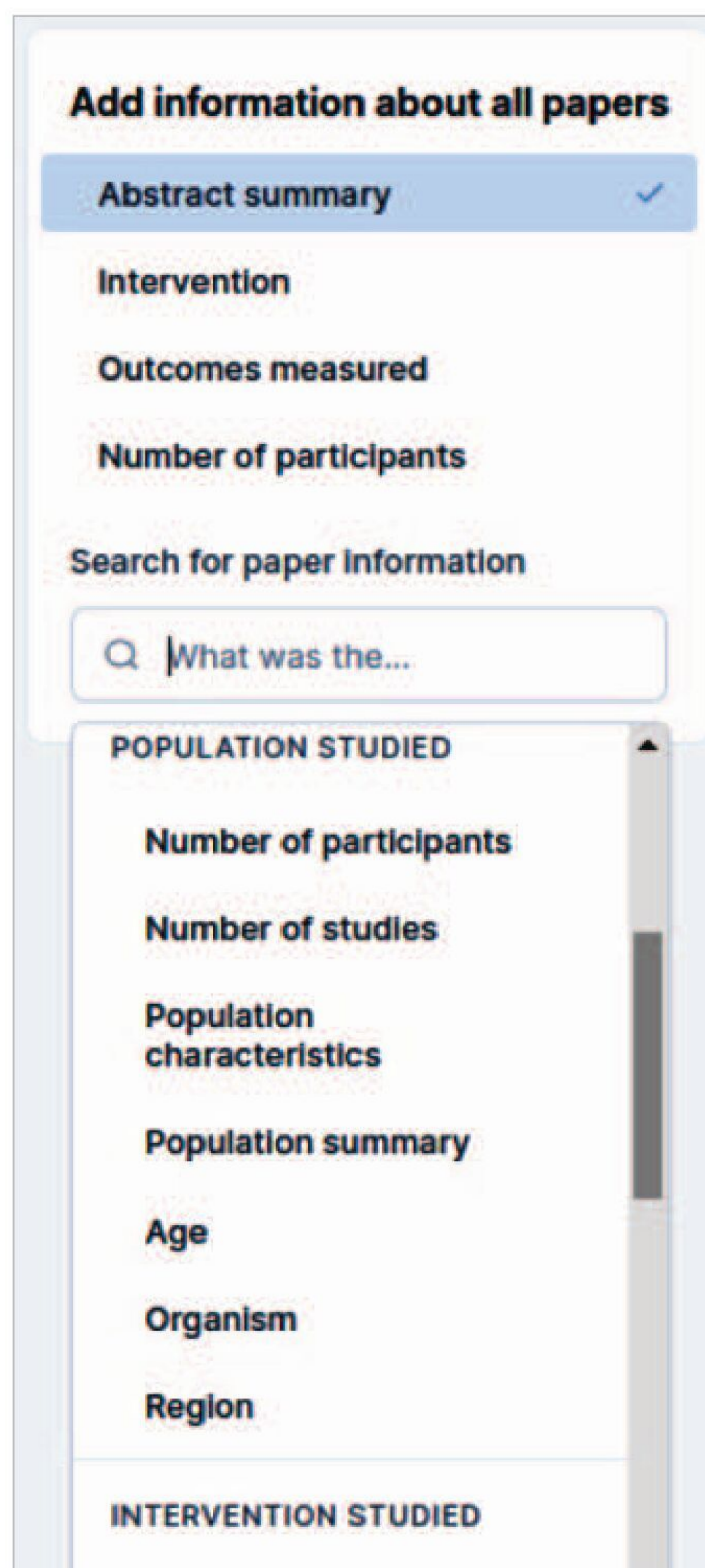


Figure 5: Add information about all papers

as Zotero are a great tool for research scholars. The results retrieved by Elicit can be exported into various formats such as CSV or .bib (bibliography). These files can be imported into reference managers of your choice for building your repository.

Let's have a simple demo of Elicit. You can access Elicit without requiring to install anything in your system by navigating to <https://elicit.org/>.

The interface is very simple and intuitive. You can simply ask a research question to Elicit and it will return the result page. In the Elicit search box, I entered the question "What is the impact of Covid-19 on primary school education in India?" Elicit conducted the search and returned the results as shown in Figure 4.

In Figure 4, notice the summary of the abstract. It facilitates scholars in deciding which papers match better with their current requirement. A salient feature of Elicit is that it enables you to add the following information about the papers in the result list:

- Abstract summary (already added by default)
- Intervention
- Outcomes measured
- Number of participants

You can sort by any field, as mentioned earlier. In Figure 6, the results are sorted by *Citation* in a decreasing order.

The results can be exported as a .bib file or CSV file as shown in Figure 7.

Once you click a particular result, the individual paper's details will be listed, which will include the following:

- Abstract summary
- What outcomes did they measure
- Who were the participants
- Can I trust this paper?
- Possible critiques
- Other citations

One more interesting option that Elicit provides is the ability to ask *Custom Questions*. It finds the answers and displays the results, as shown in Figure 9.

If you are interested in learning more about how Elicit works, navigate to <https://elicit.org/faq#appendix-how-does-elicit-work>.

Remember that Elicit works on the basis of language models and

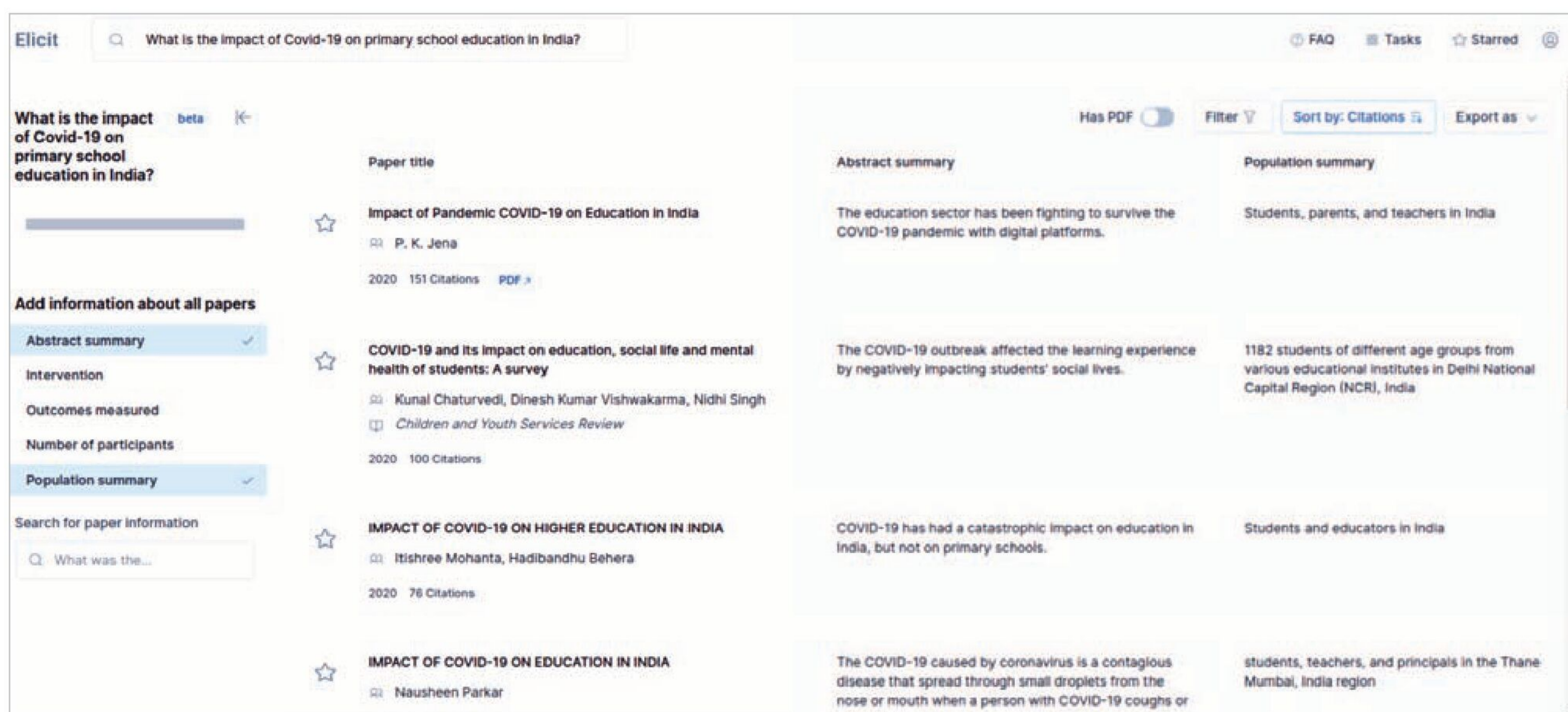


Figure 6: Elicit results sorted by *Citation*

PIONEERS IN TECHNOLOGY. PARTNERS IN SUCCESS.

EFYGROUP
YOURS SINCE 1969

**"It has helped me
reach this position."**

**-Mukesh Ajwani
MD, Electronic Components
Trading Firm**



Mr Ajwani is not the only one who owes his success to *Electronics For You*. The technology magazine that started off in 1969 is today, South Asia's most popular one. Covering the latest on emerging technologies, changing industry trends, evolving trade practices...

Not surprisingly, top decision makers and those high up on the corporate ladder in the electronics fraternity treat EFY as their Bible.

- Over half a million readers
- India's largest selling technology magazine
- Caters to the entire electronics fraternity

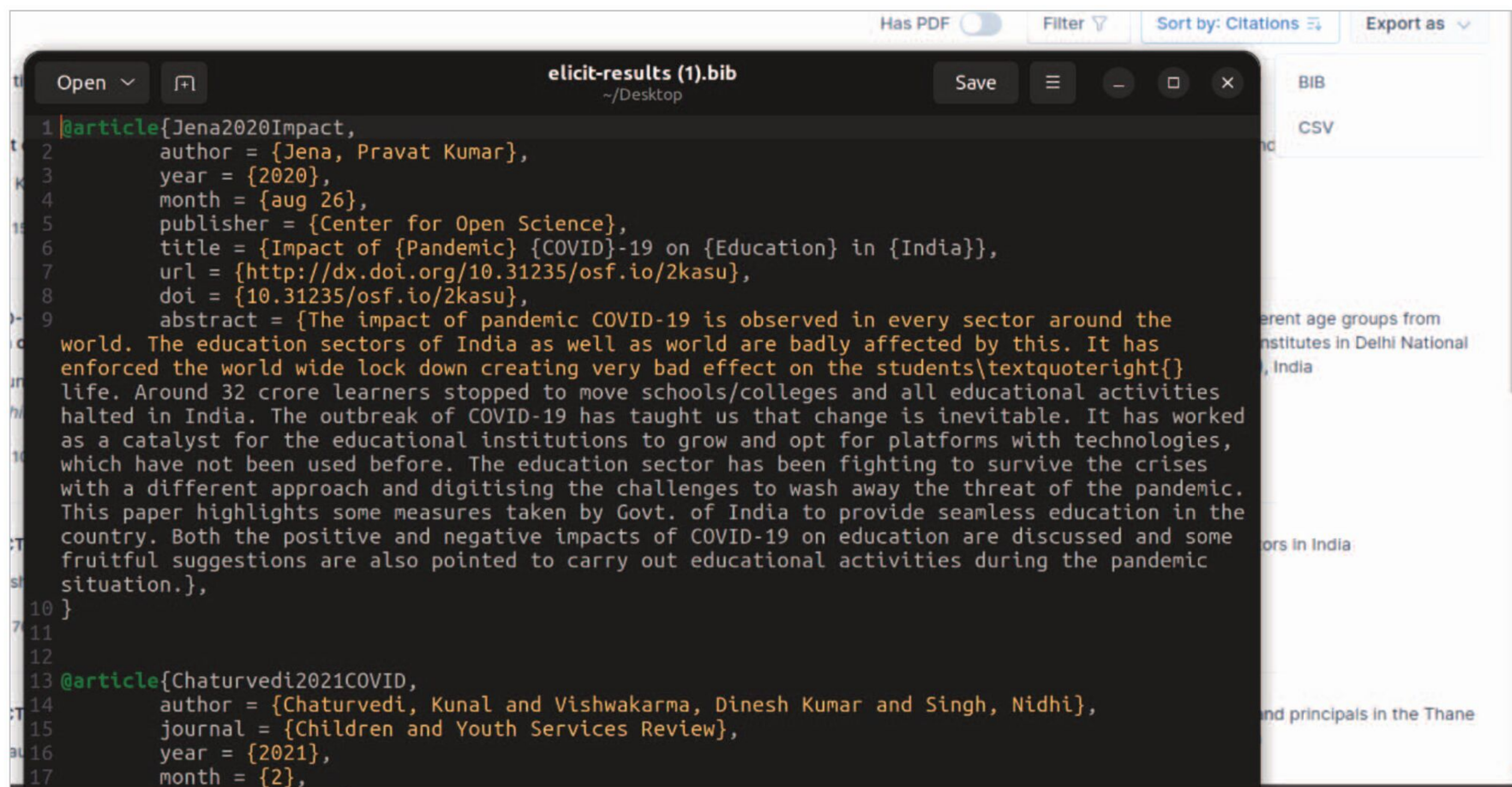


Figure 7: Elicit export result list as .bib file

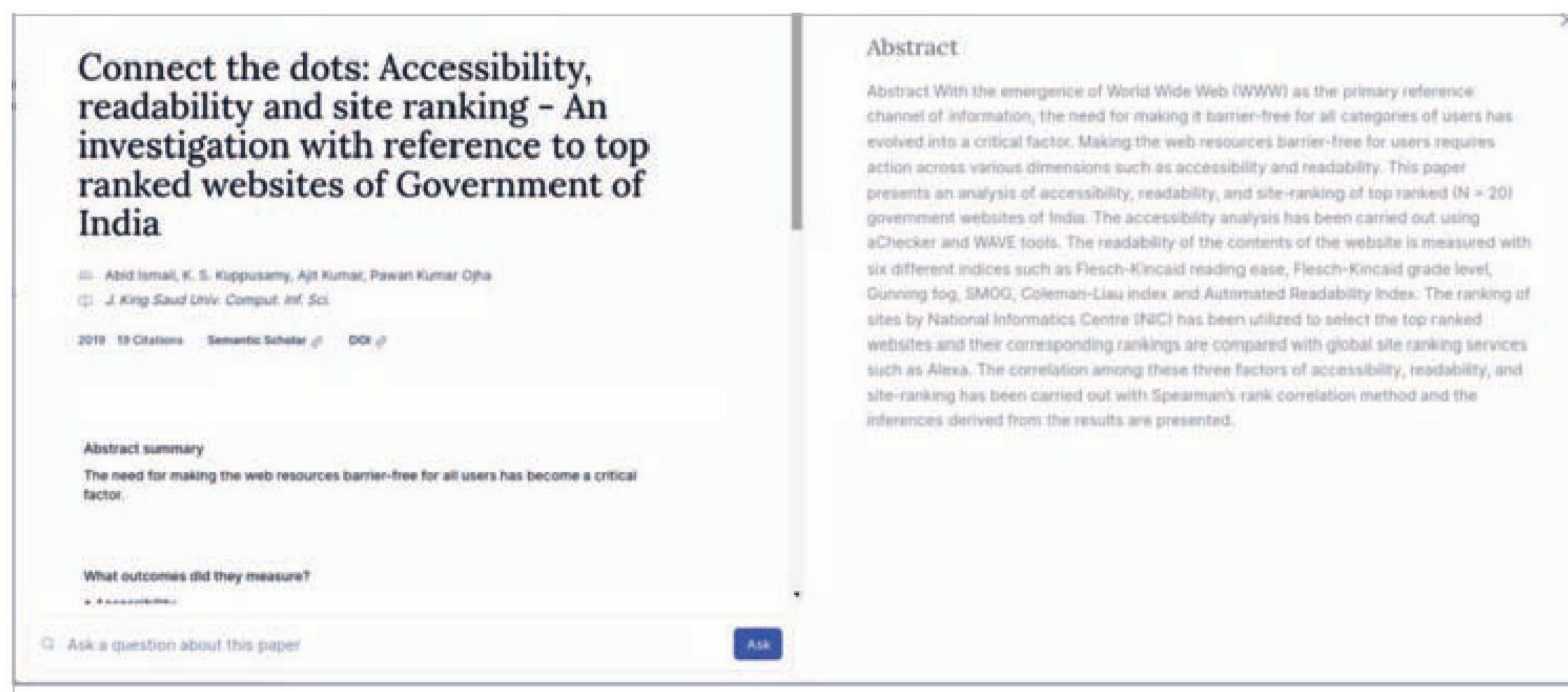


Figure 8: Elicit – Individual paper result details

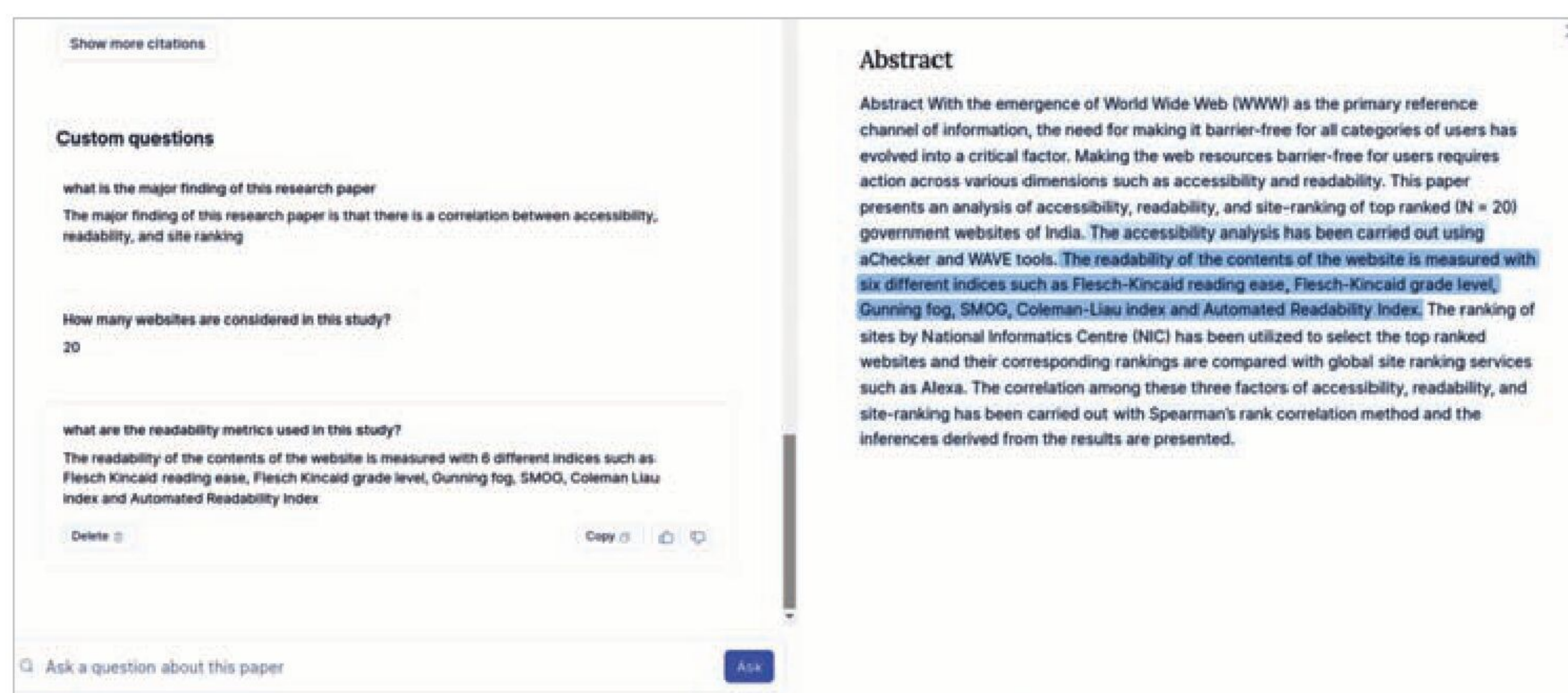


Figure 9: Elicit Custom Questions

the tool is in an early stage of development. It is the responsibility of the scholar to validate the results. However, Elicit is a great tool for researchers.

Explainpaper

Elicit helps you find relevant literature documents. The next challenge is to understand the papers. Another web based tool called Explainpaper helps with this.

Explainpaper enables the scholar to understand the contents of a research paper. The workflow is very simple here:

- Upload a paper
- Highlight a confusing text
- Get an explanation

In the home page, you can simply click on *Upload Paper* and select the PDF of a research paper. The Explainpaper interface will be ready in a few seconds. In this interface you can

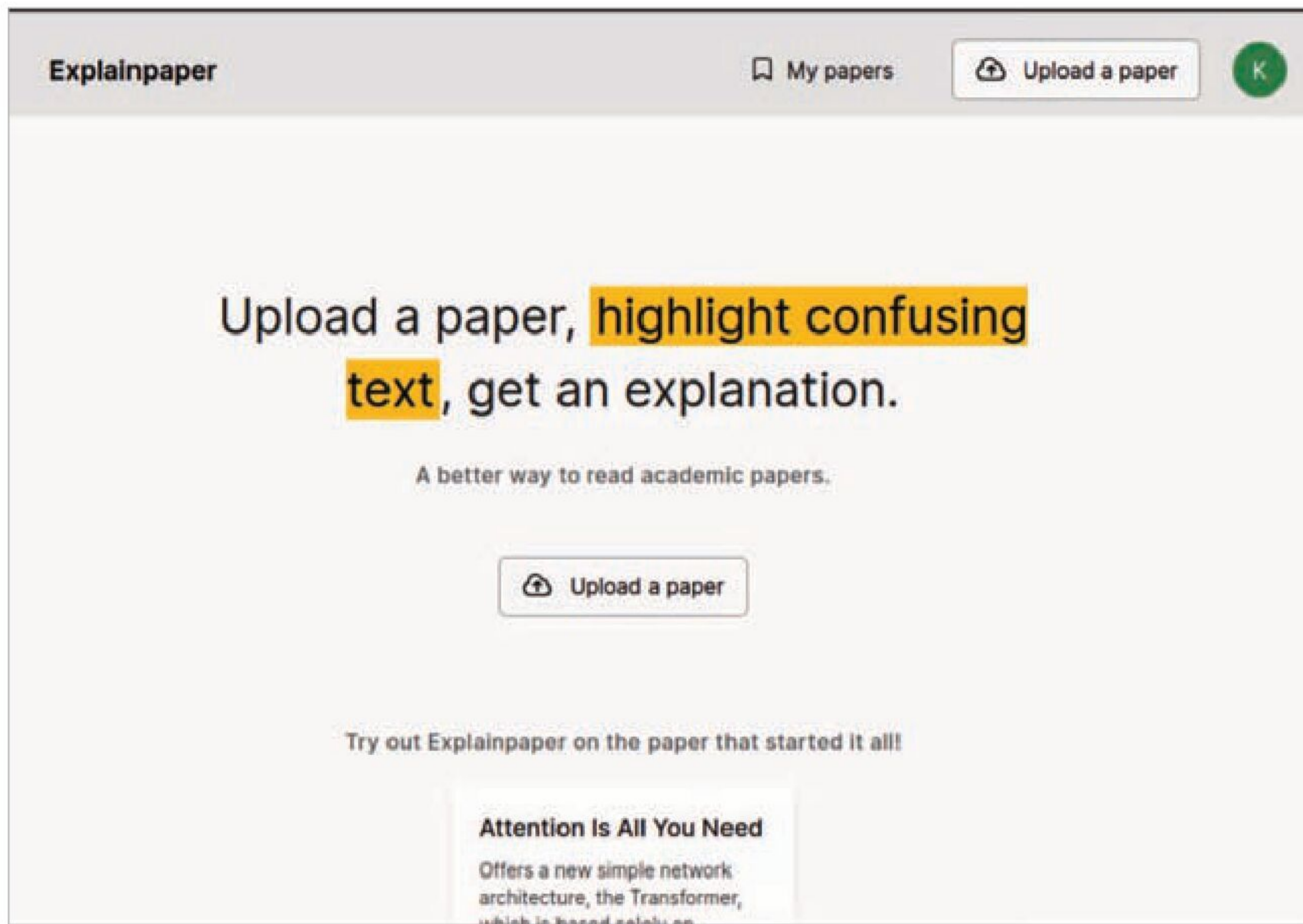


Figure 10: Explainpaper home page

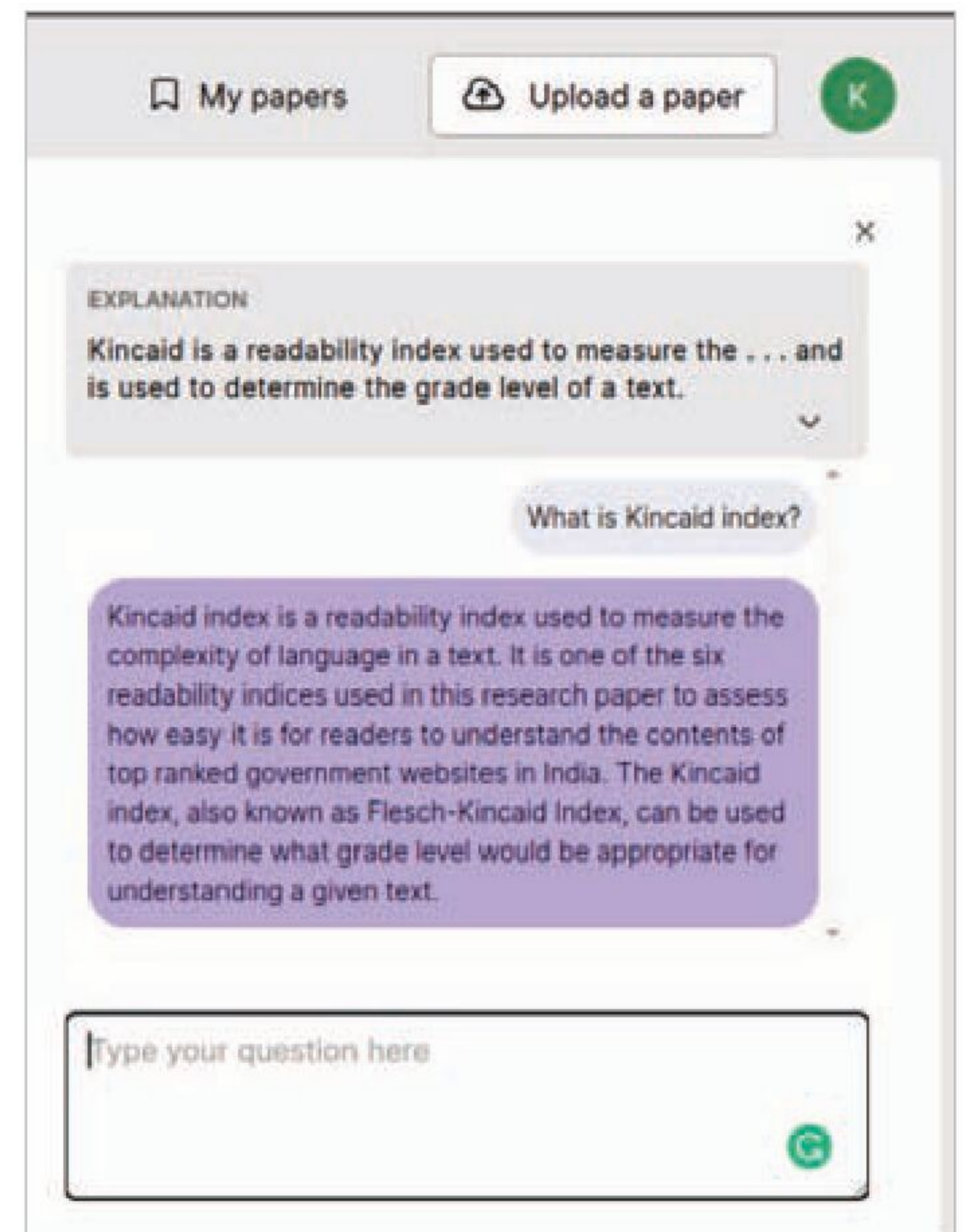


Figure 12: Explainpaper -- Follow up question

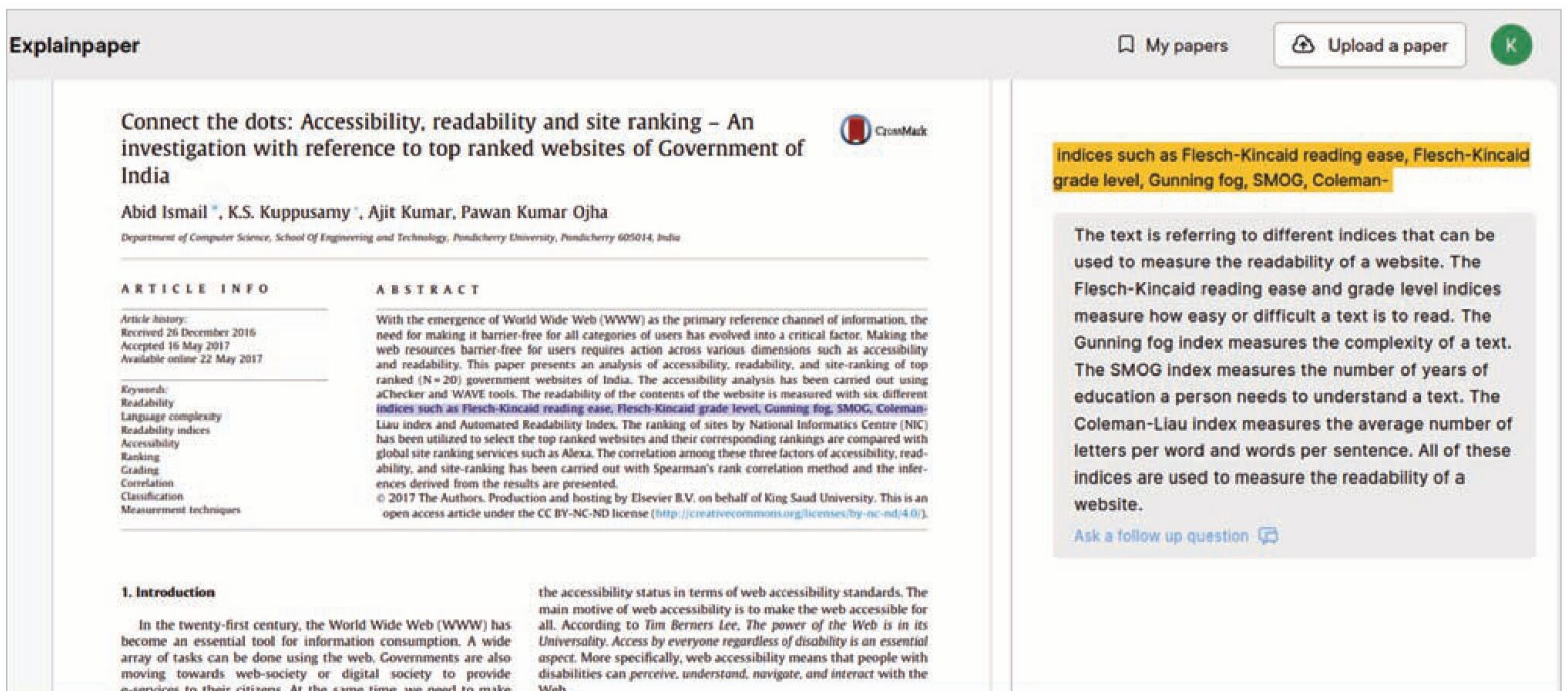


Figure 11: Explainpaper – Sample paper interaction

highlight any text and get explanations for it in simple to understand terms, as shown in Figure 11.

The best part here is that it allows you to ask follow-up questions. The screenshot in Figure 12 demonstrates a follow up question and the answer rendered by Explainpaper.

Explainpaper is also based on GPT-3 and the initial reviews are really good. It has the potential

to become a must-have tool in the near future.

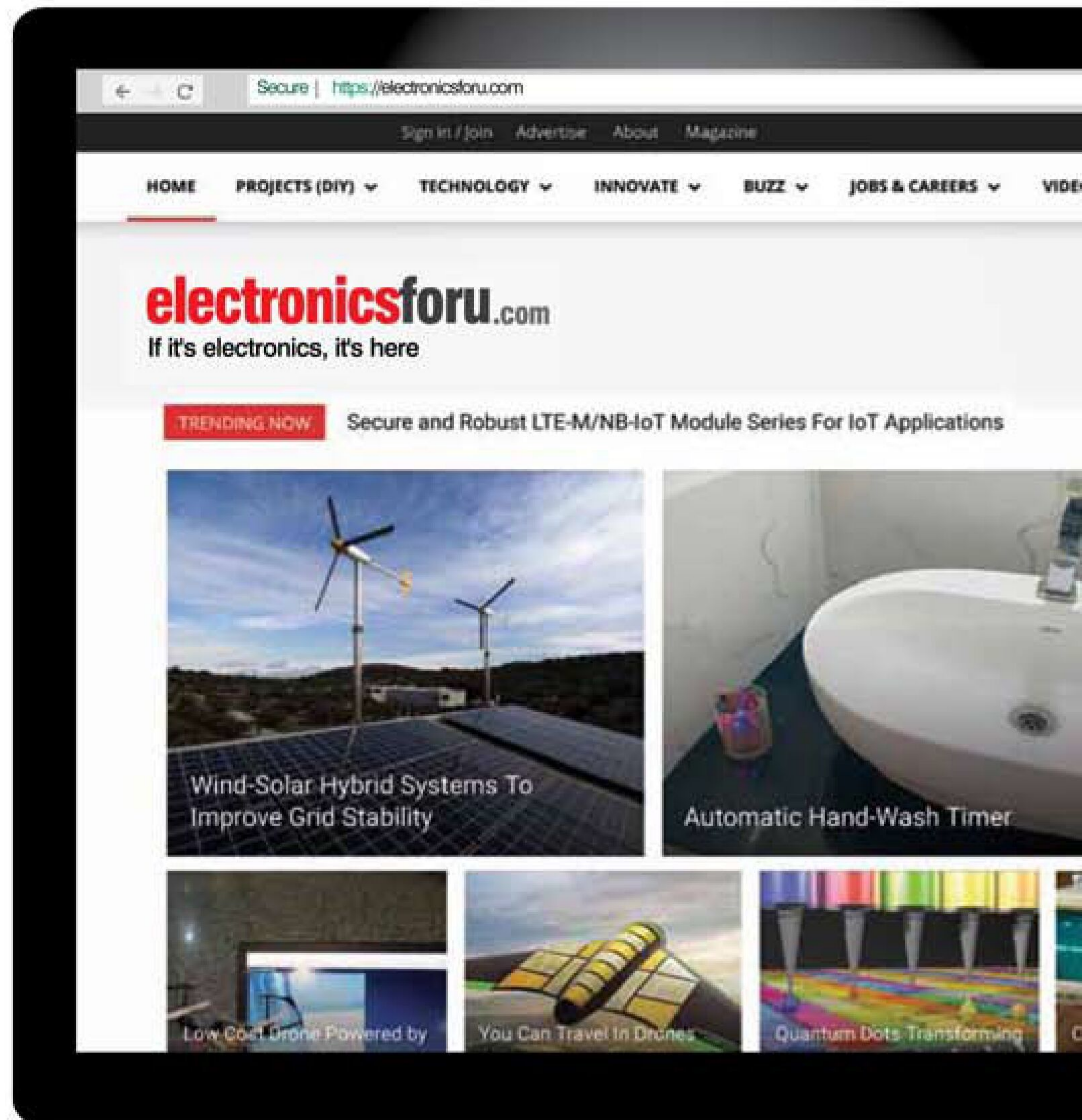
Both Elicit and Explainpaper provide powerful features. They help research scholars to navigate

literature effectively and understand papers easily. The friendly and natural interaction both these tools offer makes us feel like the future is already here! **END** 🐧

By: Dr K.S. Kuppusamy

The author is associate professor of Computer Science, School of Engineering and Technology, Pondicherry Central University, Pondicherry. He has 16+ years of teaching and research experience in academia and industry. His research interests include accessible computing, web information retrieval, and mobile computing.

Your favourite website has



electronicsforu.com

THANKS TO YOU—OUR ONLINE NETWORK IS

FACTS & FIGURES

- 4 websites (two more coming soon)
- Five major Facebook communities
- Seven major LinkedIn groups & pages
- Million-plus active users (monthly)
- Million-plus reach through Facebook
- Fifty-thousand-plus industry connections through LinkedIn

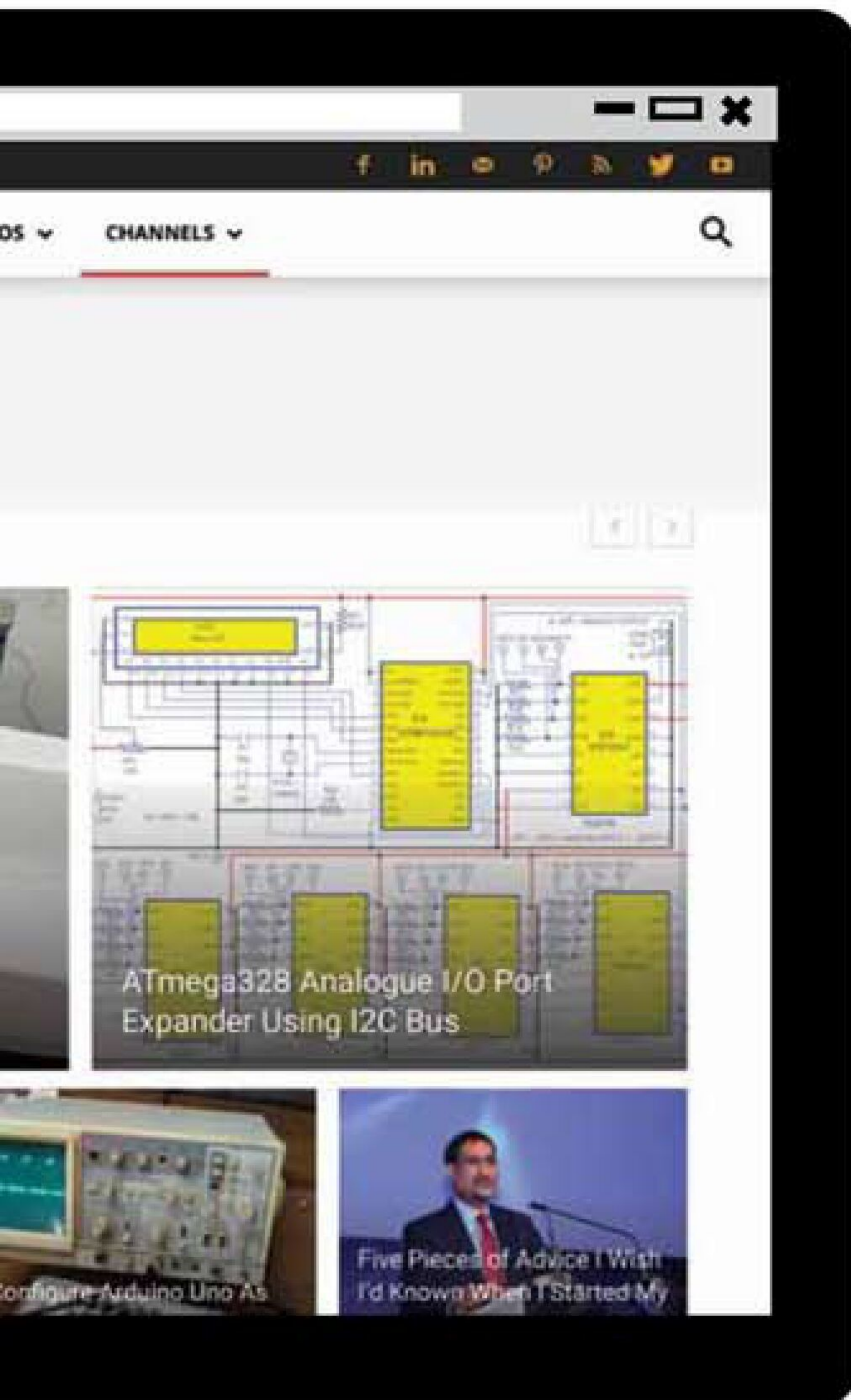
READERS

- You can access all content for FREE
- You can subscribe to newsletters for FREE --on most websites
- Register on our websites to get free invites to technical webinars and seminars

EXPERTS

- Experts who want to share their knowledge through articles, DIY Projects, etc are welcome
- We also welcome experts who want to share their knowledge through webinars or seminars
- You can contact us at editop@efy.in

fast growing peers now...



Amazing DIY Projects. Latest Tech trends.
The hang-out for electronics enthusiasts.



The Latest in IOT.
A platform for enablers, creators and providers of IOT solutions.



India. Electronics. Directory.
Enabling commerce between buyers & sellers of electronics in India.



Business. Electronics. India.
Everything you want to know about India's electronics industry.

AMONGST THE WORLD'S TOP 5 AND GROWING!

INDUSTRY

- You can advertise for as little as US\$ 100 per month
- Special combo offers for advertisers in our print publications
- We've now enabled flexible CPM-based advertising
- You can advertise on the platform of your choice (based on your target audience)
- We invite press releases at efy-edit-team@efy.in
- Press releases are published free of cost, subject to discretion of the editorial team

RESPONSE GUARANTEED SOLUTIONS

We now also act as marketing partners for our clients and drive entire marketing for them, where we charge them on basis of results and not efforts!

CONTACT US: Shrikant Rao • growmybiz@efy.in • +91-98111 55335

The Best Way to Learn Container and Cloud Native Technologies

Companies are today looking for people skilled in container and cloud native technologies. In case you want to acquire these skills, make sure you do so the right way.



The 10th Annual Open Source Job Report received from the Linux Foundation (<https://www.linuxfoundation.org/research/the-10th-annual-open-source-jobs-report>) makes it evident that container and cloud native skills are the most in demand (Figure 1). Let's first try to understand the scope of technologies covered under the umbrella term cloud native.

The Cloud Native Computing Foundation (CNCf) defines cloud native computing as the use of open source software, as well as technologies such as containers, microservices and service mesh, to develop and deploy scalable applications on cloud computing platforms.

The topics covered under cloud native are hence very vast, including the knowledge of different cloud computing platforms like AWS, Azure, Google Cloud, etc. If you want a list of the different tools and projects that are used under the hood of cloud native deployment, you can refer to <https://landscape.cncf.io/card-mode?project=hosted>.

There is no way one can master all of these. Such a breadth of topics overwhelms the learner, and creates confusion on where and how to begin as well. At the same time, if the basics are right, learning these topics does not take much time.

So, before learning about containers and cloud native technologies, one needs to make sure one's foundation is very strong. By this I mean one should have a sound understanding of the following:

- Operating system and networking concepts
- Working knowledge of Linux
- Working knowledge of Git and GitHub
- General understanding of cloud and DevOps

One cannot skip or jump directly into advanced topics. The topics listed above are generally taught in colleges these days, but if you have not learned them earlier, there are many resources available — both online and offline.

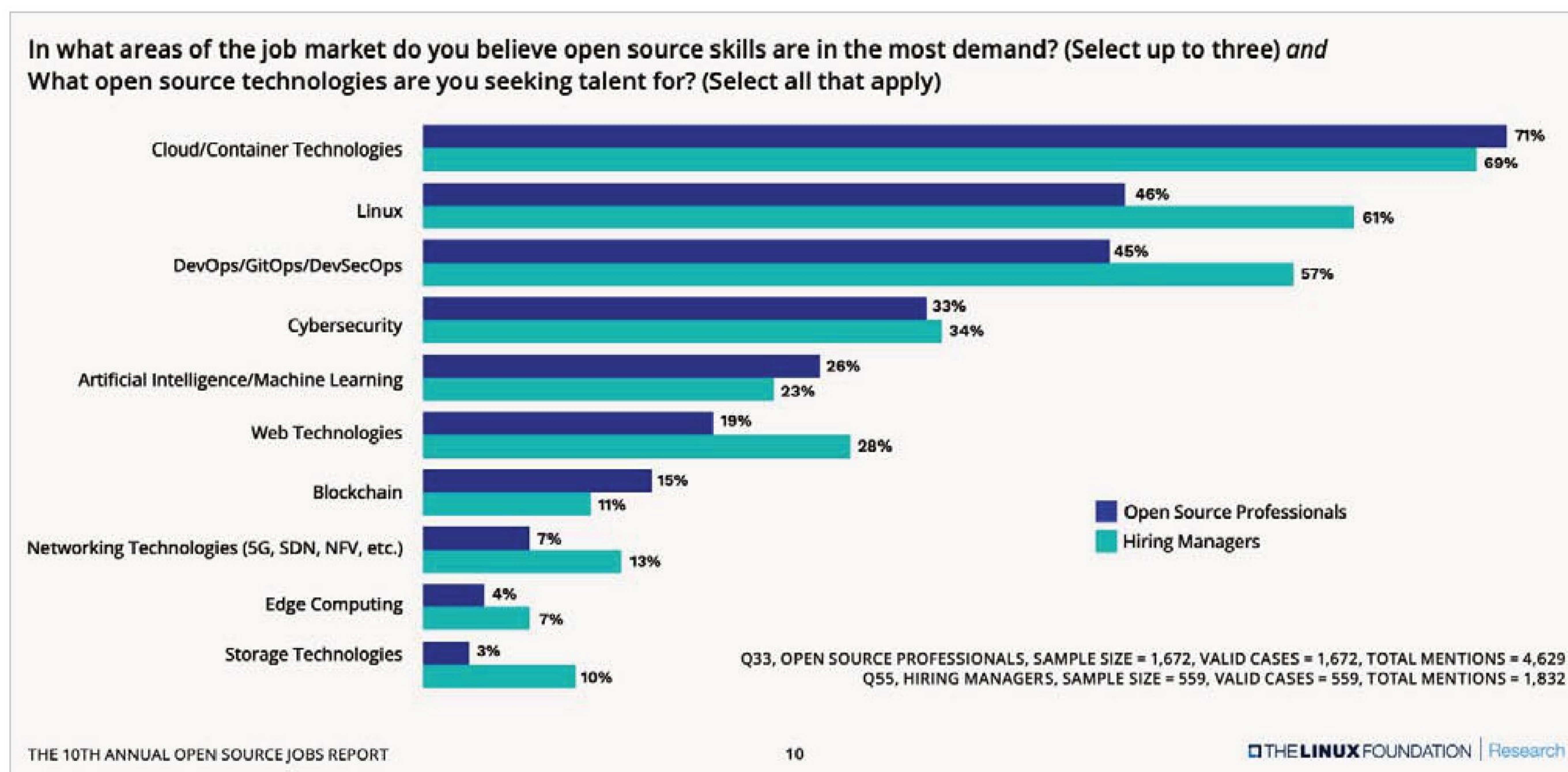


Figure 1: Technologies where open source skills are in demand

Once you have mastered these subjects, you are ready to learn container and cloud native technologies. As these cover many topics, I would like to specifically call out the topics you must learn in order to become a cloud native engineer:

- Container runtimes like *containerd*, Docker
- Kubernetes
- Helm
- Working knowledge of any public cloud (Azure, AWS or Google Cloud)
- Prometheus
- OpenTelemetry
- Overview of service mesh

Every other topic after this is an add-on, to be learnt on a need basis.

Now let us take a look at what are the different ways in which you can learn these subjects.

YouTube

Almost everyone goes to YouTube first to learn about any new topic. But this comes with its own challenges:

- A lot to choose from, which is very confusing.
- There is no defined path, and one can move in endless cycles to watch some random videos.
- Content is not up-to-date.
- There is no mentor or instructor available to guide.
- There are no hands-on labs available to practice while watching the videos.

In my opinion YouTube is not for serious learners. One can get a very good overview of a subject, which may be sufficient in a few cases.

Self-paced online training

Self-paced online training gives you the flexibility to go over the content anytime you wish to. Most such online courses are only video based but some of them now come with an in-built lab environment. So choose the latter, whenever possible.

When you start looking for courses, I would suggest exploring official free training, if available. CNCF has an introductory course on Kubernetes on Edx called ‘Introduction to Kubernetes’ (<https://www.edx.org/course/introduction-to-kubernetes>). Just for your information, I authored its initial version.

Once you have completed the official free courses, you can explore other courses from different online training websites, which offer some great content (refer to the box).

List of online resources for learning cloud native technologies

- <https://training.linuxfoundation.org>
- <https://cloudyuga.guru>
- <https://kodekloud.com>
- <https://www.pluralsight.com>
- <https://cloudacademy.com>
- <https://learnk8s.io/academy>
- <https://www.udemy.com>

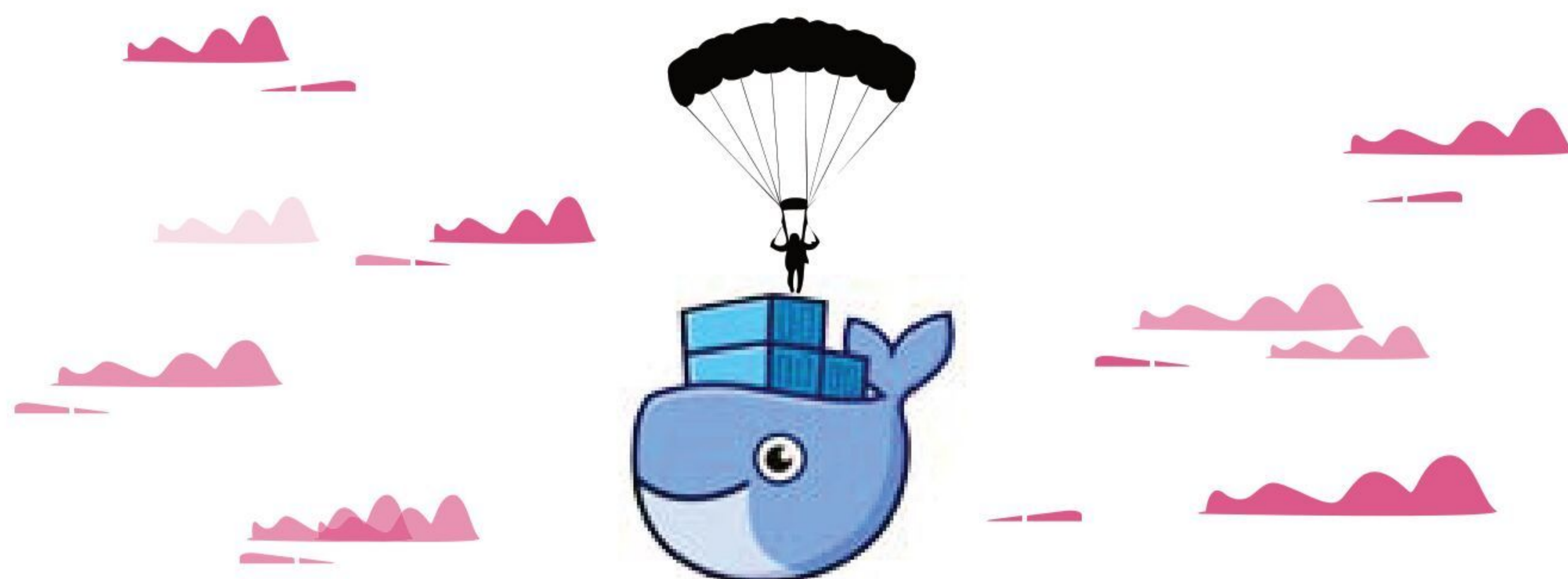
Self-paced mode of learning requires a lot of internal motivation and discipline to finish the courses, as learners are on their own. Active participation in local community events like meetups, workshops and becoming a part of the online communities is highly advisable in case you are pursuing this kind of training.

Continued to page no...43

FOCUS

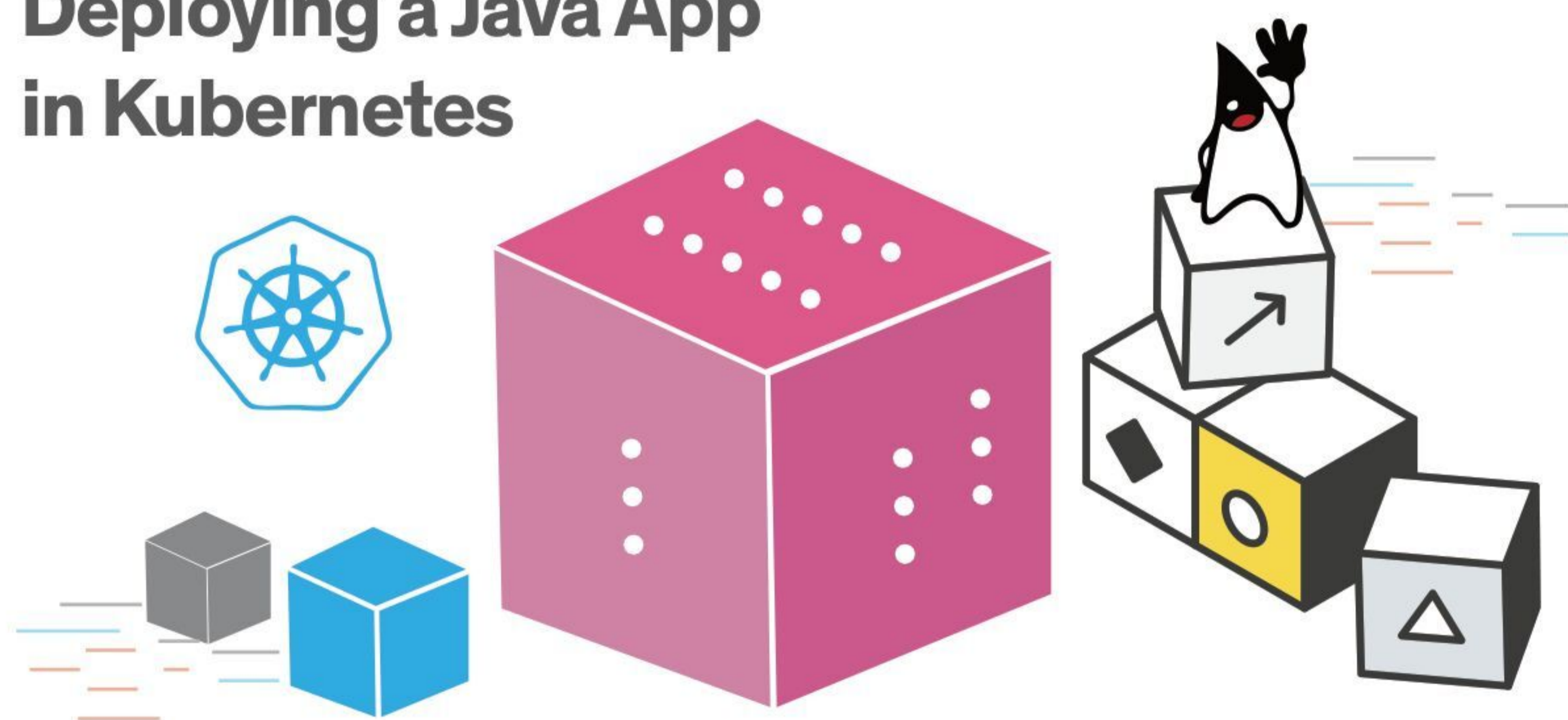
Exploring Containers

Getting Started with Docker



Docker is a well-known term today, and we often hear of it in relation to DevOps. This article is for those of you who want to learn about this amazing virtualisation software but don't know where to begin. It will tell you how to install and start using it. [Read more on page.....41](#)

Deploying a Java App in Kubernetes



This article will tell you how to install minikube and use it for deploying a sample application. It will also tell you how to build a Docker image and run it. [Read more on page.....50](#)

Docker SBOM: Keeping a Check on Security Risks



The Log4j vulnerability and SolarWinds supply chain attack have made us realise that software supply chains are at great risk of being targeted by attackers. The Docker SBOM helps detect these risks.

[Read more on page.....65](#)

Free Tools and Libraries that Help to Keep Docker and Containers Secure



Let's take a quick look at the free tools and libraries that are available for the security of Docker and containers.

[Read more on page.....62](#)

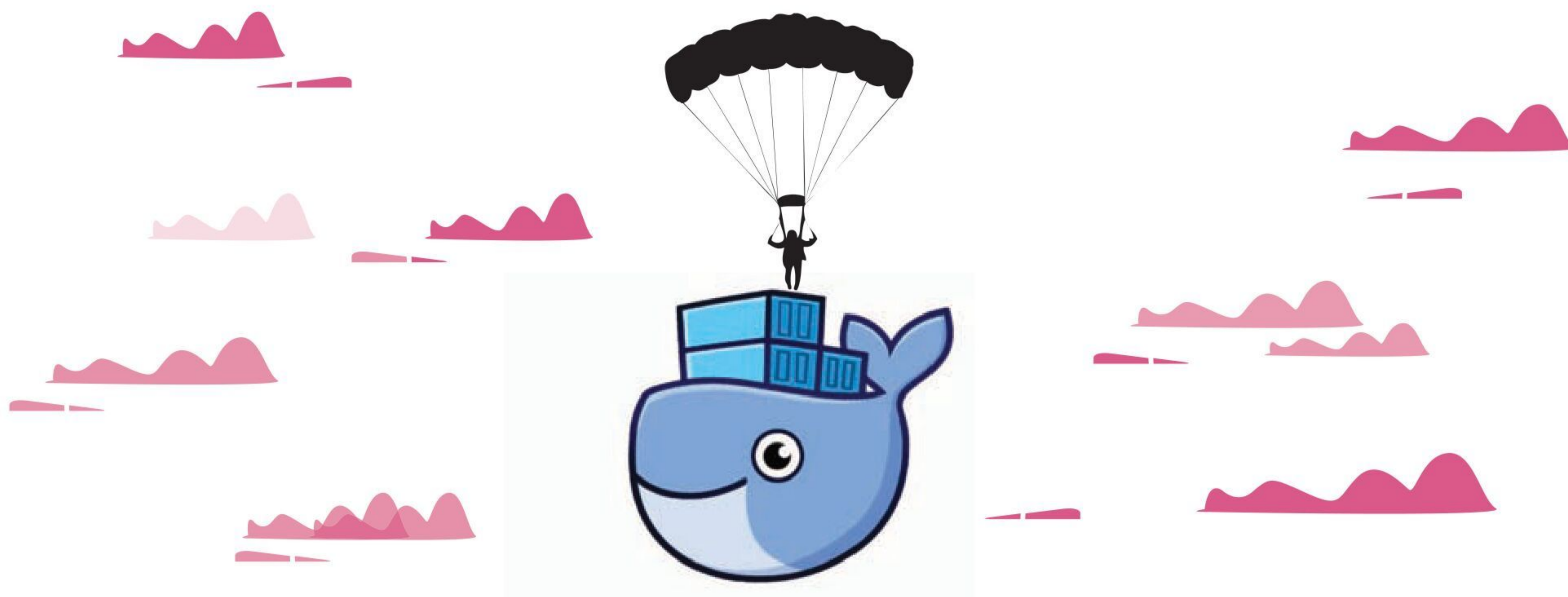
Podman: The Pod Manager Tool



Podman (POD manager) is a daemon-less and open source Linux native tool by Red Hat for developing, managing, and running containers. It deploys applications using the Open Containers Initiative (OCI) container and container Images. Podman manages pods, containers, container images, etc, using the libpod library. Let's learn a few more things about it.

[Read more on page.....73](#)

Getting Started with Docker



Docker is a well-known term today, and we often hear of it in relation to DevOps. This article is for those of you who want to learn about this amazing virtualisation software but don't know where to begin. It will tell you how to install and start using it.

Docker is basically an operating system level virtualisation software that is used to provide software applications in a file in the form of containers. Docker Inc., which developed this software, offers both paid and free services.

This application based virtualisation software is really important today, because unlike virtual machines it does not need much space or processing power. It only needs the Docker software to run the container, and the container therefore becomes compatible with any operating system or environment. Almost all IT companies today use Docker for one service or another, making it an essential skill for software engineers and open source enthusiasts.

So let's learn how to install and set up Docker on our Ubuntu machines, and then we will run our

```

jishnusaaurav@jishnusaaurav-L340-15IRH-GAMING:~$ sudo apt update
[sudo] password for jishnusaaurav:
Get:1 http://packages.microsoft.com/repos/code stable InRelease [10.4 kB]
Get:2 https://dl.google.com/linux/chrome/deb stable InRelease [1,811 B]
Get:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:4 http://packages.microsoft.com/repos/code stable/main arm64 Packages [123 kB]
Hit:5 https://packages.microsoft.com/repos/ms-teams stable InRelease
Get:6 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,127 B]
Hit:7 http://in.archive.ubuntu.com/ubuntu bionic InRelease
Get:8 http://packages.microsoft.com/repos/code stable/main amd64 Packages [122 kB]
Hit:9 http://ppa.launchpad.net/obsproject/obs-studio/ubuntu bionic InRelease
Get:10 http://packages.microsoft.com/repos/code stable/main armhf Packages [122 kB]
Get:11 http://in.archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Hit:12 http://ppa.launchpad.net/strukturag/libheif/ubuntu bionic InRelease
Hit:13 http://ppa.launchpad.net/uroni/urbackup/ubuntu bionic InRelease
Get:14 http://in.archive.ubuntu.com/ubuntu bionic-backports InRelease [83.3 kB]
Get:15 http://security.ubuntu.com/ubuntu bionic-security/main i386 Packages [1,278 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu bionic-updates/main i386 Packages [1,572 kB]
Get:17 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease [3,626 B]
Get:18 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [2,810 kB]
Get:19 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [2,472 kB]
Get:20 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [428 kB]
Get:21 http://in.archive.ubuntu.com/ubuntu bionic-updates/main Translation-en [517 kB]
Get:22 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 DEP-11 Metadata [297 kB]
Get:23 http://security.ubuntu.com/ubuntu bionic-security/main amd64 DEP-11 Metadata [55.2 kB]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 DEP-11 Metadata [61.1 kB]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 DEP-11 Metadata [2,464 B]
Get:26 http://in.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 DEP-11 Metadata [302 kB]
Get:27 http://in.archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 DEP-11 Metadata [2,468 B]
Get:28 http://in.archive.ubuntu.com/ubuntu bionic-backports/main amd64 DEP-11 Metadata [8,092 B]
Get:29 http://in.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 DEP-11 Metadata [10.0 kB]
Fetched 10.5 MB in 9s (1,176 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
25 packages can be upgraded. Run 'apt list --upgradable' to see them.
jishnusaaurav@jishnusaaurav-L340-15IRH-GAMING:~$

```

Figure 1: Package manager update

first Docker container, which is like the ‘hello-world’ of the Docker/DevOps world; the container app is also called Hello-World!

To begin, update your package manager using the following command (as shown in Figure 1):

```
sudo apt update
```

Then, upgrade it to add any new packages required using the

following command:

```
sudo apt upgrade
```

The next step is to install the *docker.io*; you can use the following command (as shown in Figure 3):

```
sudo apt install docker.io
```

Now you can check if the installation worked — you can use

the following command to check the version of Docker installed:

```
jishnusaurav@jishnusaurav-L340-15IRH-GAMING:~$ docker --version
```

```
Docker version 20.10.12, build 20.10.12-0ubuntu2-18.04.1
```

You have finished installing Docker and the setup is complete. You can now pull the first image and run it

```
jishnusaurav@jishnusaurav-L340-15IRH-GAMING:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
augeas-lenses cdb5 cpu-checker cryptsetup cryptsetup-bin db-util db5.3-util dh-translations dmeventd extlinux hfsplus lbverbs-providers icoutils intltool ipxe-qemu ipxe-qemu-256k-compat-efi-roms jq
ldntool libafflib0v5 libaio1 libaugeas0 libbft01 libcacard0 libconfig9 libdate-manip-perl libdevmapper-event1.02.1 libewf2 libfdt1 libfile-which-perl libguestfs-hfsplus libguestfs-perl
libguestfs-reiserfs libguestfs-tools libguestfs-xfs libguestfs0 libhfs0 libhivex0 libibverbs1 libintl-perl libintl-xs-perl libiscsi7 libjq1 libldm-1.0-0 liblvm2app2.2 liblvm2cmd2.02 libnl-route-3-200
libonig4 librados2 librbdi1 librdnsc1 libspice-server1 libstring-shellquote-perl libsys-virt-perl libtsk13 libusbredirparser1 libvirt0 libwin-hivex-perl libxen-4.9 libxenstore3.0
linux-hwe-5.4-headers-5.4.0-100 linux-hwe-5.4-headers-5.4.0-104 linux-hwe-5.4-headers-5.4.0-105 linux-hwe-5.4-headers-5.4.0-107 linux-hwe-5.4-headers-5.4.0-109 linux-hwe-5.4-headers-5.4.0-117
linux-hwe-5.4-headers-5.4.0-121 linux-hwe-5.4-headers-5.4.0-122 linux-hwe-5.4-headers-5.4.0-124 linux-hwe-5.4-headers-5.4.0-125 linux-hwe-5.4-headers-5.4.0-128 linux-hwe-5.4-headers-5.4.0-42
linux-hwe-5.4-headers-5.4.0-99 lsscsi lvm2 nss-tools osinfo-db python3-scour qemu-block-extra qemu-system-common qemu-system-x86 qemu-utils scour scrub seabios sgabios sleuthkit sqlite3 supermin
zerofree
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
libpcre2-16-0 libpcre2-32-0 libpcre2-dev libpcre2-posix0
The following packages have been kept back:
libheif-examples
The following packages will be upgraded:
code distro-info-data gir1.2-notify-0.7 google-chrome-stable grub-efi-amd64-bin grub-efi-amd64-signed klibc-utils libc-bin libice-dev libice6 libklibc libnotify-bin libnotify4 libsensors4 locales
login multirch-support passwd r-base-dev r-doc-html snapd teams ubuntu-advantage-tools
23 upgraded, 4 newly installed, 0 to remove and 1 not upgraded.
Need to get 322 MB of archives.
After this operation, 86.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://packages.microsoft.com/repos/code stable/main amd64 code amd64 1.73.1-1667967334 [96.7 MB]
Get:2 https://dl.google.com/linux/chrome/deb stable/main amd64 google-chrome-stable amd64 107.0.5304.121-1 [93.3 MB]
Get:3 https://packages.microsoft.com/repos/ms-teams stable/main amd64 teams amd64 1.5.00.23801 [85.5 MB]
Get:4 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ r-base-dev 4.2.2-20221110-1.1804.0 [4,492 B]
Get:5 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 login amd64 1:4.5-1ubuntu2.4 [307 kB]
Get:6 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ r-doc-html 4.2.2-20221110-1.1804.0 [587 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 locales all 2.27-3ubuntu1.6 [3,613 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libc-bin amd64 2.27-3ubuntu1.6 [640 kB]
Get:9 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 passwd amd64 1:4.5-1ubuntu2.4 [817 kB]
Get:10 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 distro-info-data all 0.37ubuntu0.15 [4,724 B]
Get:11 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 klibc-utils amd64 2.0.4-9ubuntu2.2 [110 kB]
Get:12 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libklibc amd64 2.0.4-9ubuntu2.2 [41.8 kB]
Get:13 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 ubuntu-advantage-tools amd64 27.11.3-18.04.1 [161 kB]
Get:14 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 multirch-support amd64 2.27-3ubuntu1.6 [0,960 B]
Get:15 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libnotify4 amd64 0.7.7-3ubuntu0.18.04.1 [18.7 kB]
Get:16 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 gir1.2-notify-0.7 amd64 0.7.7-3ubuntu0.18.04.1 [3,588 B]
Get:17 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 grub-efi-amd64-signed amd64 1.173.2-18.04.1+2.04-1ubuntu47.4 [403 kB]
Get:18 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 grub-efi-amd64-bin amd64 2.04-1ubuntu47.4 [730 kB]
Get:19 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libice-dev amd64 2:1.0.9-2ubuntu0.18.04.1 [47.2 kB]
Get:20 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libice6 amd64 2:1.0.9-2ubuntu0.18.04.1 [40.7 kB]
Get:21 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libnotify-bin amd64 0.7.7-3ubuntu0.18.04.1 [6,784 B]
Get:22 http://in.archive.ubuntu.com/ubuntu bionic/universe amd64 libpcre2-16-0 amd64 10.31-2 [169 kB]
Get:23 http://in.archive.ubuntu.com/ubuntu bionic/universe amd64 libpcre2-32-0 amd64 10.31-2 [160 kB]
Get:24 http://in.archive.ubuntu.com/ubuntu bionic/universe amd64 libpcre2-posix0 amd64 10.31-2 [5,788 B]
Get:25 http://in.archive.ubuntu.com/ubuntu bionic/universe amd64 libpcre2-dev amd64 10.31-2 [627 kB]
Get:26 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libsensors4 amd64 1:3.4.0-4ubuntu0.1 [28.3 kB]
Get:27 http://in.archive.ubuntu.com/ubuntu bionic-updates/main amd64 snapd amd64 2.57.5+18.04 [37.6 MB]
26% [27 snapd 5,045 kB/37.6 MB 13%] [2 google-chrome-stable 17.5 MB/93.3 MB 19%] [1 code 3,097 kB/96.7 MB 3%] [3 teams 808 kB/85.5 MB 1%] 3,256 kB/s In[28]
```

Figure 2: Package manager upgrade

```
jishnusaurav@jishnusaurav-L340-15IRH-GAMING:~$ sudo apt install docker.io
[sudo] password for jishnusaurav:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
augeas-lenses cdb5 cpu-checker cryptsetup cryptsetup-bin db-util db5.3-util dh-translations dmeventd extlinux hfsplus lbverbs-providers icoutils intltool ipxe-qemu ipxe-qemu-256k-compat-efi-roms jq
ldntool libafflib0v5 libaio1 libaugeas0 libbft01 libcacard0 libconfig9 libdate-manip-perl libdevmapper-event1.02.1 libewf2 libfdt1 libfile-which-perl libguestfs-hfsplus libguestfs-perl
libguestfs-reiserfs libguestfs-tools libguestfs-xfs libguestfs0 libhfs0 libhivex0 libibverbs1 libintl-perl libintl-xs-perl libiscsi7 libjq1 libldm-1.0-0 liblvm2app2.2 liblvm2cmd2.02 libnl-route-3-200
libonig4 librados2 librbdi1 librdnsc1 libspice-server1 libstring-shellquote-perl libsys-virt-perl libtsk13 libusbredirparser1 libvirt0 libwin-hivex-perl libxen-4.9 libxenstore3.0
linux-hwe-5.4-headers-5.4.0-100 linux-hwe-5.4-headers-5.4.0-104 linux-hwe-5.4-headers-5.4.0-105 linux-hwe-5.4-headers-5.4.0-107 linux-hwe-5.4-headers-5.4.0-109 linux-hwe-5.4-headers-5.4.0-117
linux-hwe-5.4-headers-5.4.0-121 linux-hwe-5.4-headers-5.4.0-122 linux-hwe-5.4-headers-5.4.0-124 linux-hwe-5.4-headers-5.4.0-125 linux-hwe-5.4-headers-5.4.0-128 linux-hwe-5.4-headers-5.4.0-42
linux-hwe-5.4-headers-5.4.0-99 lsscsi lvm2 nss-tools osinfo-db python3-scour qemu-block-extra qemu-system-common qemu-system-x86 qemu-utils scour scrub seabios sgabios sleuthkit sqlite3 supermin
zerofree
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan
Suggested packages:
aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
bridge-utils containerd docker.io git git-man liberror-perl pigz runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 2 not upgraded.
Need to get 73.6 MB of archives.
After this operation, 367 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu bionic/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu bionic/main amd64 bridge-utils amd64 1.5-15ubuntu1 [30.1 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 runc amd64 1.1.0-0ubuntu1-18.04.1 [3,896 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 containerd amd64 1.5.9-0ubuntu1-18.04.1 [33.0 MB]
39% [4 containerd 25.8 MB/33.0 MB 78%] 2,021 kB/s 21s
```

Figure 3: Installation

```

jishnusaurav@jishnusaurav-L340-15IRH-GAMING:~$ docker run hello-world
docker: Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/containers/create": dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
jishnusaurav@jishnusaurav-L340-15IRH-GAMING:~$ clear
jishnusaurav@jishnusaurav-L340-15IRH-GAMING:~$ sudo docker run hello-world
[sudo] password for jishnusaurav:
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:faa03e786c97f07ef34423fccceec2398ec8a5759259f94d99078f264e9d7af
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

jishnusaurav@jishnusaurav-L340-15IRH-GAMING:~$

```


Figure 4: Using the *hello-world* image on Docker

(as shown in Figure 4).

`sudo docker run hello-world`

So, that is all! You have run your first Docker container! That is pretty

cool. Now go ahead and read the Docker documentation, which will help

you use more and better application containers on Docker. **END** 

 **By: Jishnu Saurav Mittapalli**

The author is a full stack Web developer, interested in modern technologies like artificial intelligence and machine learning. He is currently working in the field of natural language processing.

Continued from page no...39

Instructor-led online training

Time bound learning opportunities are the best as you can complete them in a dedicated time frame with the help of an instructor, using hands-on labs. These trainings are scheduled more often than physical ones, as people can join them remotely as well.

You may not understand something fully during the sessions, but there is always the content/recording to refer to, in order to connect the dots.

Websites that offer instructor-led online training

- <https://training.linuxfoundation.org>
- <https://cloudyuga.guru>
- <https://learnk8s.io/academy>
- <https://www.edureka.co/>
- <https://www.simplilearn.com>

Make sure you look at the trainer profile before attending such training, as a good trainer can make things very simple to understand.

Offline training

This kind of training is the most expensive, but should not be missed if you happen to find it. Nothing can substitute

live interaction. Companies can schedule such training easily for their employees.

It's advisable to choose companies or institutes that specialise in cloud native and DevOps training rather than generic ones. Also, do check for the industry experience of the trainer before signing up for courses.

A note on advanced topics

For advanced topics like setting up a hybrid Kubernetes cluster, multi-cluster monitoring, security, etc, one should get trained only by a practitioner or professional trainer.

One need not pay from one's own pocket for advanced training, as this can be quite expensive, unless there is a real use case for which it is needed. In that case, your organisation will be happy to sponsor it.

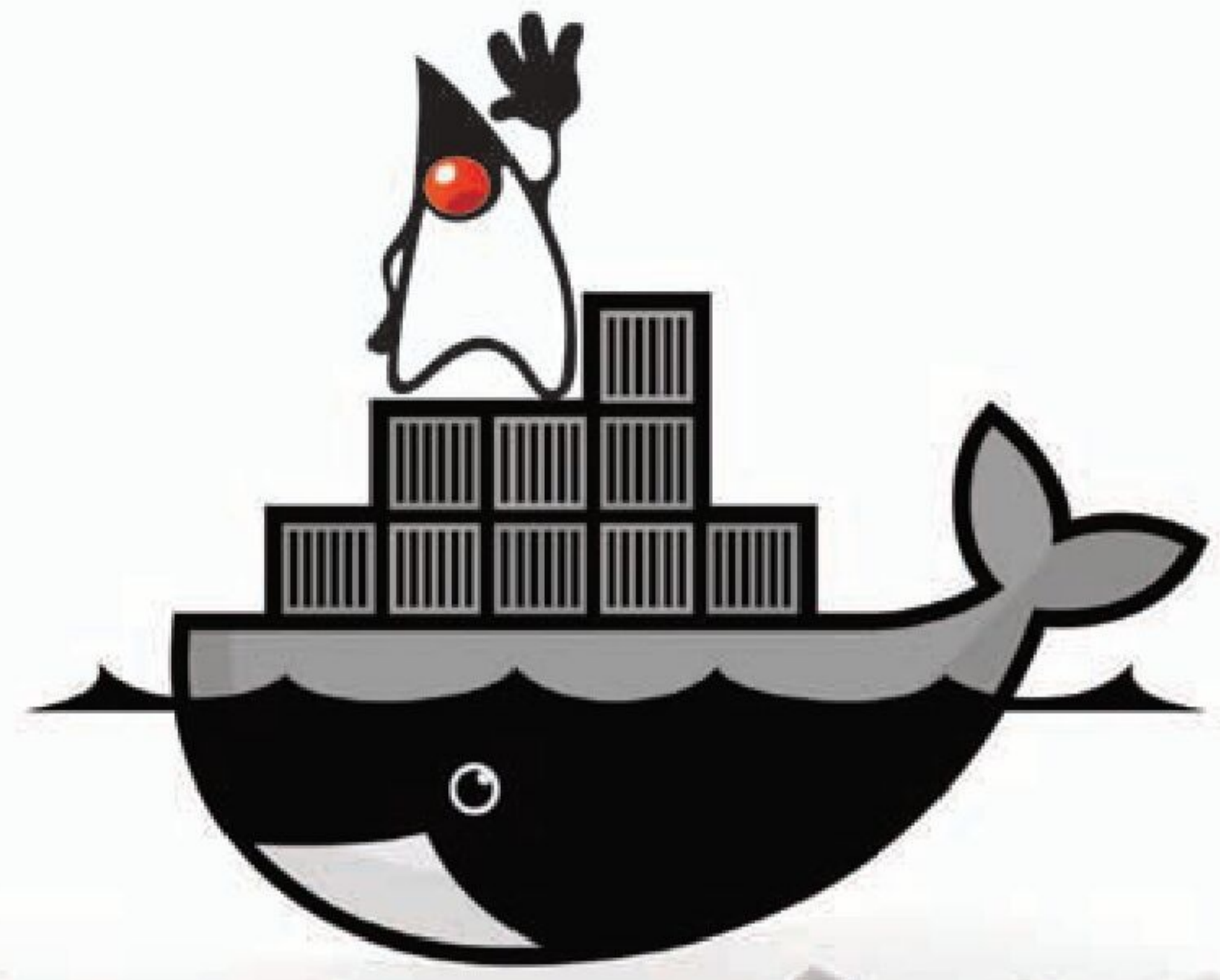
In a nutshell, one must first get the basics right and then go for the advanced topics by choosing the right training programme and method. **END** 

 **By: Neependra Khare**

The author is the founder of and principal trainer at CloudYuga. He is a published author and has created many online courses on containers and cloud native technologies.

Deploying a Java App in Docker

This article is a 'how to' on installing Docker on the Ubuntu operating system and then deploying a Java app.



Docker is a platform that helps you build, test, and deploy applications quickly, which makes development efforts efficient. It provides UIs, CLIs, APIs and security features that are very useful to make standardised components called containers. Containers have all the resources to run an application such as libraries, system tools, code, and runtime. Docker is very useful as it gives a way to quickly deploy applications into any environment. In this article we will deploy a popular Java based application in Docker. But before we do that, we will install Docker in the Ubuntu operating system.

To do that, make sure you have the following:

Ubuntu versions	Architectures
<ul style="list-style-type: none"> • Ubuntu Kinetic 22.10 • Ubuntu Jammy 22.04 (LTS) • Ubuntu Focal 20.04 (LTS) • Ubuntu Bionic 18.04 (LTS) 	<ul style="list-style-type: none"> • x86_64 (or amd64) • armhf • arm64 • s390x

Now go to *Settings* and click on the *About* tab to verify the Ubuntu version (Figure 1).



Figure 1: Details of the operating system being used

Docker installation on Ubuntu 22.04.1 LTS

First, update the *apt* package index and install packages using the following command:

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ sudo apt-get update
```

```
[sudo] password for osfy:
```

```
Ign:1 https://pkg.jenkins.io/debian-stable binary/ InRelease
```

```
Hit:2 https://pkg.jenkins.io/debian-stable binary/ Release
```

```
Hit:3 https://dl.yarnpkg.com/debian stable InRelease
```

```
Hit:4 https://dl.google.com/linux/chrome/deb stable InRelease
```

```
Hit:5 http://in.archive.ubuntu.com/ubuntu jammy InRelease
```

```
Get:8 http://in.archive.ubuntu.com/ubuntu jammy-updates
```

```
InRelease [114 kB]
```

```
Get:6 https://packages.cloud.google.com/apt kubernetes-xenial
```

```
InRelease [9,383 B]
```

```
Get:9 http://security.ubuntu.com/ubuntu jammy-security
```

```
InRelease [110 kB]
```

```
Hit:10 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu
```

```
jammy InRelease
```

```
Get:11 http://in.archive.ubuntu.com/ubuntu jammy-backports
```

```
InRelease [99.8 kB]
```

```
Hit:12 https://deb.nodesource.com/node_18.x jammy InRelease
```

```
Get:13 https://ppa.launchpadcontent.net/openshot.developers/
```

```
libopenshot-daily/ubuntu jammy InRelease [17.6 kB]
```

```
Get:14 http://in.archive.ubuntu.com/ubuntu jammy-updates/main
```

```
i386 Packages [385 kB]
```

```
Get:15 http://in.archive.ubuntu.com/ubuntu jammy-updates/main
```

```
amd64 Packages [736 kB]
```

```
.
```

```
.
```

```
.
```

```
Get:28 http://in.archive.ubuntu.com/ubuntu jammy-backports/
```

```
universe Translation-en [9,460 B]
```

```
Get:29 http://in.archive.ubuntu.com/ubuntu jammy-backports/
```

```
universe amd64 DEP-11 Metadata [11.7 kB]
```

```

.
Fetches 3,548 kB in 6s (585 kB/s)

```

```

Reading package lists... Done

```

```

W: https://dl.yarnpkg.com/debian/dists/stable/InRelease: Key
is stored in legacy trusted.gpg keyring (/etc/apt/trusted.
gpg), see the DEPRECATION section in apt-key(8) for details.

```

Next, we will install the packages:

```

osfy@ubuntu-22-04-1-lts:~/Desktop$ sudo apt-get install ca-
certificates curl gnupg lsb-release

```

Now add Docker's official GPG key:

```

osfy@ubuntu-22-04-1-lts:~/Desktop$ sudo mkdir -p /etc/apt/
keyrings
osfy@ubuntu-22-04-1-lts:~/Desktop$ curl -fsSL https://
download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg
File '/etc/apt/keyrings/docker.gpg' exists. Overwrite? (y/N)
Y

```

After this, set up the repository:

```

osfy@ubuntu-22-04-1-lts:~/Desktop$ echo "deb [arch=$(dpkg
--print-architecture) signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu $(lsb_release -cs)
stable" | sudo tee /etc/apt/sources.list.d/docker.list > /
dev/null
osfy@ubuntu-22-04-1-lts:~/Desktop$

```

Next, let's update the *apt* package index:

```

osfy@ubuntu-22-04-1-lts:~/Desktop$ sudo apt-get update

```

It is now time to install the latest version of Docker in Ubuntu 22.04.1 LTS:

```

osfy@ubuntu-22-04-1-lts:~/Desktop$ sudo apt-get install
docker-ce docker-ce-cli containerd.io docker-compose-plugin

```

Let's verify the Docker version using the *sudo docker version* command:

```

osfy@ubuntu-22-04-1-lts:~/Desktop$ sudo docker version

```

```

Client: Docker Engine - Community

```

```

Version:      20.10.21
API version:  1.41
Go version:   go1.18.7
Git commit:   baeda1f
Built:        Tue Oct 25 18:01:58 2022

```

```

OS/Arch:      linux/amd64
Context:      default
Experimental: true

```

```

Server: Docker Engine - Community

```

```

Engine:
Version:      20.10.21
API version:  1.41 (minimum version 1.12)
Go version:   go1.18.7
Git commit:   3056208
Built:        Tue Oct 25 17:59:49 2022
OS/Arch:      linux/amd64
Experimental: false
containerd:
Version:      1.6.10
GitCommit:    770bd0108c32f3fb5c73ae1264f7e503fe7b2661
runc:
Version:      1.1.4
GitCommit:    v1.1.4-0-g5fd4c4d
docker-init:
Version:      0.19.0
GitCommit:    de40ad0

```

Now let's verify the status of the Docker service:

```

osfy@ubuntu-22-04-1-lts:~/Desktop$ service docker status
× docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service;
   enabled; vendor preset: enabled)
   Active: failed (Result: exit-code) since Thu 2022-12-01
00:05:56 IST; 26s ago
   TriggeredBy: × docker.socket
   Docs: https://docs.docker.com
   Process: 51801 ExecStart=/usr/bin/dockerd -H fd://
--containerd=/run/containerd/containerd.sock (code=exited,
status=1/FAILURE)
   Main PID: 51801 (code=exited, status=1/FAILURE)
   CPU: 206ms

```

Since the status of the Docker service is not successful, stop the service and start again:

```

osfy@ubuntu-22-04-1-lts:~/Desktop$ systemctl stop docker.
service
osfy@ubuntu-22-04-1-lts:~/Desktop$ systemctl start docker.
service
osfy@ubuntu-22-04-1-lts:~/Desktop$ systemctl status docker.
service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service;
   enabled; vendor preset: enabled)
   Active: active (running) since Thu 2022-12-01 00:07:13

```

```

IST; 3s ago
TriggeredBy: ● docker.socket
  Docs: https://docs.docker.com
Main PID: 51913 (dockerd)
  Tasks: 17
  Memory: 26.5M
  CPU: 894ms
  CGroup: /system.slice/docker.service
          └─51913 /usr/bin/dockerd -H fd:// --containerd=
run/containerd/containerd.sock

```

We now have Docker installed and running successfully on Ubuntu 22.04.1 LTS. So let's verify the number of Docker images available. We already have minikube installed and configured; so one image is available. You may find this list empty in your case.

```

osfy@ubuntu-22-04-1-lts:~/Desktop$ sudo docker images

REPOSITORY              TAG          IMAGE ID
CREATED                SIZE
gcr.io/k8s-minikube/kicbase  v0.0.35    7fb60d0ea30e   8
weeks ago             1.12GB

```

Verify the number of Docker containers available. You may find this list empty in your case.

```

osfy@ubuntu-22-04-1-lts:~/Desktop$ sudo docker ps -a

CONTAINER ID   IMAGE                                COMMAND
CREATED        STATUS          PORTS          NAMES
bddcebbb9e3e   gcr.io/k8s-minikube/kicbase:v0.0.35  "/usr/
local/bin/entr..." 7 weeks ago   Exited (137) 6 days ago
minikube

```

Let's verify Docker installation by running a container. Run *nginx* in the Docker container using the *docker run* command. In case the image is not available locally, it will be downloaded.

```

osfy@ubuntu-22-04-1-lts:~/Desktop$ sudo docker run -d -p
9999:80 nginx

```

```

Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
a603fa5e3b41: Pull complete
c39e1cda007e: Pull complete
90cfefba34d7: Pull complete
a38226fb7aba: Pull complete
62583498bae6: Pull complete
9802a2cfdb8d: Pull complete
Digest: sha256:e209ac2f37c70c1e0e9873a5f7231e91dcd83fdf11

```

```

78d8ed36c2ec09974210ba
Status: Downloaded newer image for nginx:latest

```

```

a23fc125bbca2650c6b9a2a15787dfe9312b97a01c2894b7bed0
5453870339d2

```

Now, verify the Docker images and run containers again. We can find nginx images and containers in a list that was not available earlier.

```

osfy@ubuntu-22-04-1-lts:~/Desktop$ sudo docker images

```

```

REPOSITORY              TAG          IMAGE ID
CREATED                SIZE
nginx                   latest       88736fe82739   2
weeks ago             142MB
gcr.io/k8s-minikube/kicbase  v0.0.35    7fb60d0ea30e   8
weeks ago             1.12GB

```

```

osfy@ubuntu-22-04-1-lts:~/Desktop$ sudo docker ps -a

```

```

CONTAINER ID   IMAGE                                COMMAND
CREATED        STATUS          PORTS          NAMES
a23fc125bbca   nginx                                "/
docker-entrypoint..." 8 seconds ago   Up 6 seconds
0.0.0.0:9999->80/tcp, :::9999->80/tcp  festive_mclaren
bddcebbb9e3e   gcr.io/k8s-minikube/kicbase:v0.0.35  "/usr/
local/bin/entr..." 7 weeks ago   Exited (137) 6 days ago

```

Visit <http://localhost:9999/> and you will find the nginx 'Welcome' page.

We have successfully verified Docker installation. We must now do Java and Maven verification.

Java and Maven installation

We need Java and Maven to build a package file for a sample application that we are going to run in a Docker container. Install Java and Maven in Ubuntu OS. Once that's done successfully, verify the Java version using the *java -*



Figure 2: nginx home page

version command:

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ java -version

openjdk version "11.0.17" 2022-10-18
OpenJDK Runtime Environment (build 11.0.17+8-post-Ubuntu-
1ubuntu222.04)
OpenJDK 64-Bit Server VM (build 11.0.17+8-post-Ubuntu-
1ubuntu222.04, mixed mode, sharing)
```

Verify the Maven version using the `mvn -version` command:

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ mvn -version

Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 11.0.17, vendor: Ubuntu, runtime: /usr/lib/jvm/
java-11-openjdk-amd64
Default locale: en_IN, platform encoding: UTF-8
OS name: "linux", version: "5.15.0-53-generic", arch:
"amd64", family: "unix"
```

We are going to use PetClinic as a sample application. PetClinic is a Spring Boot application built using Maven or Gradle. In this article we will build a JAR file and run it from the Docker container. You can visit <https://github.com/spring-projects/spring-petclinic> to know more.

Copy the GitHub repository URL from the above URL and use the `git clone` command to clone it locally.

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ git clone https://github.
com/spring-projects/spring-petclinic.git
```

```
Cloning into 'spring-petclinic'...
remote: Enumerating objects: 9424, done.
remote: Total 9424 (delta 0), reused 0 (delta 0), pack-reused
9424
Receiving objects: 100% (9424/9424), 7.64 MiB | 1.55 MiB/s,
done.
Resolving deltas: 100% (3567/3567), done.
```

Once source code is available, go to the root of the project using the `change directory` command as given below, and execute the `./mvnw package` command to create a JAR file. It will try to download multiple dependencies and hence will take some time to complete execution.

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ cd spring-petclinic
osfy@ubuntu-22-04-1-lts:~/Desktop/spring-petclinic$ ./mvnw
package
Warning: JAVA_HOME environment variable is not set.
[INFO] Scanning for projects...
```

```
Downloading from spring-snapshots: https://repo.spring.
io/snapshot/org/springframework/boot/spring-boot-starter-
parent/2.7.3/spring-boot-starter-parent-2.7.3.pom
Downloading from spring-milestones: https://repo.spring.
io/milestone/org/springframework/boot/spring-boot-starter-
parent/2.7.3/spring-boot-starter-parent-2.7.3.pom
Downloading from central: https://repo.maven.apache.
org/maven2/org/springframework/boot/spring-boot-starter-
parent/2.7.3/spring-boot-starter-parent-2.7.3.pom
.
.
.
Downloaded from central: https://repo.maven.apache.
org/maven2/org/springframework/session/spring-session-
bom/2021.2.0/spring-session-bom-2021.2.0.pom (3.1 kB at 6.0
kB/s)
[INFO]
[INFO] -----< org.springframework.samples:spring-
petclinic >-----
[INFO] Building petclinic 2.7.3
[INFO] -----[ jar ]-----
Downloading from spring-snapshots: https://repo.spring.io/
snapshot/org/springframe
Downloaded from central: https://repo.maven.apache.org/
maven2/org/codehaus/plexus/plexus-io/3.2.0/plexus-io-
3.2.0.jar (76 kB at 28 kB/s)
Downloaded from central: https://repo.maven.apache.org/
maven2/org/codehaus/plexus/plexus-archiver/4.2.7/plexus-
archiver-4.2.7.jar (195 kB at 69 kB/s)
Downloaded from central: https://repo.maven.apache.org/
maven2/org/apache/commons/commons-compress/1.20/commons-
compress-1.20.jar (632 kB at 218 kB/s)
Downloaded from central: https://repo.maven.apache.org/
maven2/org/tukaani/xz/1.9/xz-1.9.jar (116 kB at 36 kB/s)
Downloaded from central: https://repo.maven.apache.org/
maven2/org/codehaus/plexus/plexus-utils/3.3.1/plexus-utils-
3.3.1.jar (262 kB at 70 kB/s)
[INFO] Building jar: /home/osfy/Desktop/spring-petclinic/
target/spring-petclinic-2.7.3.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.7.3:repackage
(repackage) @ spring-petclinic ---
```

Check the `.m2` folder available in the `Home` directory. It will have all the dependencies downloaded in the `mvn` package execution, as seen in Figure 3.

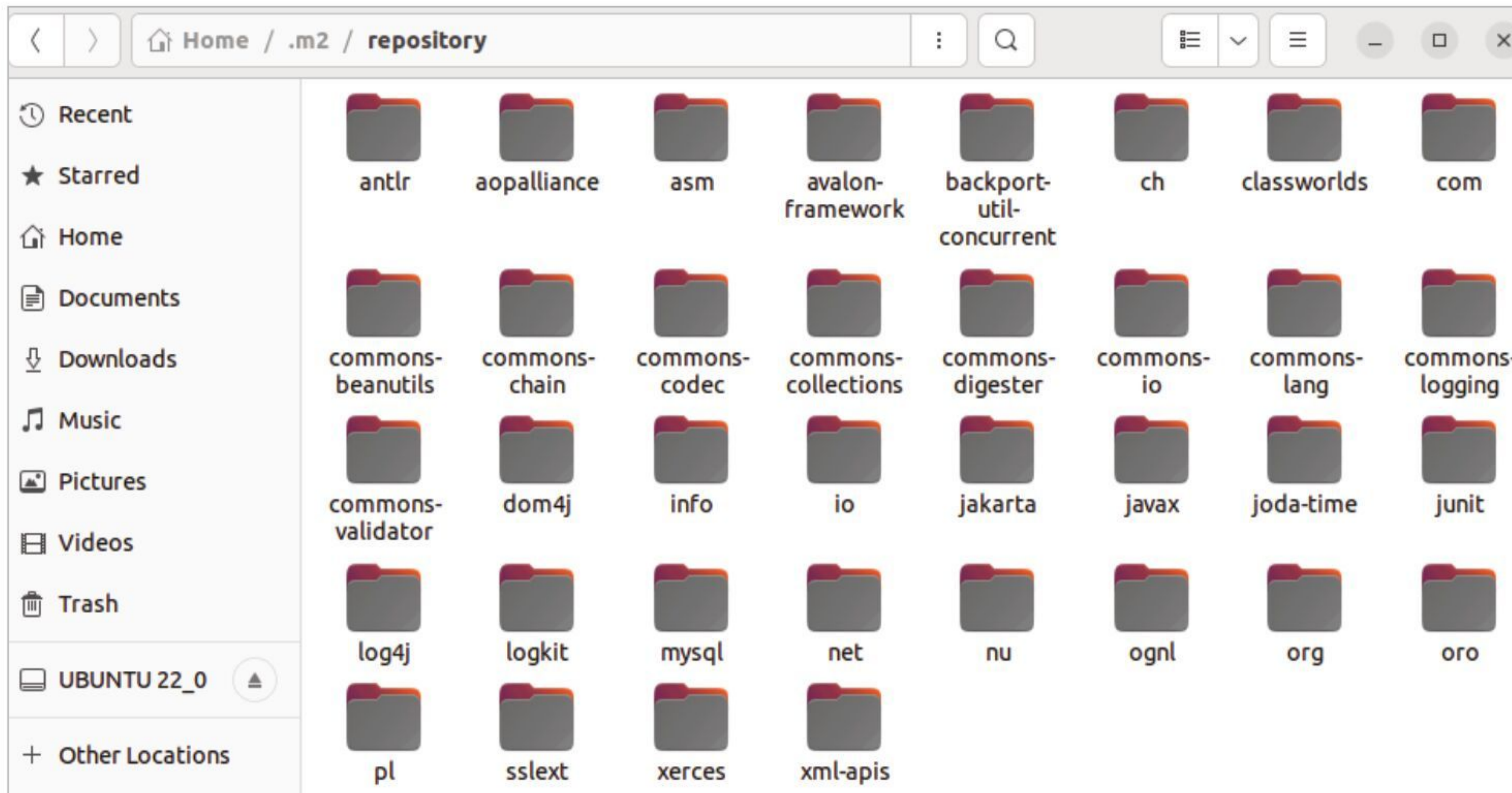


Figure 3: Folder view with all dependencies

Observe the logs of Maven command execution where it says *Building jar: /home/osfy/Desktop/spring-petclinic/target/spring-petclinic-2.7.3.jar*.

Now, we have a JAR file ready. The next step is to create a Docker image that contains the JAR file and runtime environment.

So, what is a Dockerfile? Well, it's nothing but the text document that provides build instructions to build the image with your application package.

```
FROM openjdk:11
MAINTAINER osfy
COPY target/spring-petclinic-2.7.3.jar app.jar
ENTRYPOINT ["java","-jar","/app.jar"]
```

Let's understand the above steps.

- **FROM** command: Uses the base image (openjdk:11) for subsequent instructions in Dockerfile.
- **COPY** command: Copies a file to the container.
- **ENTRYPOINT** command: Specifies entry point and manipulates the environment.

Place the Dockerfile in the *project root* directory. Now it is time to build a Docker image.

Building a Docker image and running a container

Execute the:

```
sudo docker image build -t petclinic:latest
```

...command from the *project root* directory. The output of the command will be similar to what's given below:

```
osfy@ubuntu-22-04-1-lts:~/Desktop/spring-petclinic$ sudo
docker image build -t petclinic:latest .
[sudo] password for osfy:
```

Sending build context to Docker daemon 66.47MB

```
Step 1/4 : FROM openjdk:11
11: Pulling from library/openjdk
001c52e26ad5: Pull complete
d9d4b9b6e964: Pull complete
2068746827ec: Pull complete
9daef329d350: Pull complete
d85151f15b66: Pull complete
66223a710990: Pull complete
db38d58ec8ab: Pull complete
Digest: sha256:99bac5bf83633e3c7399aed725c8415e7b569b54e03e4599e580fc9cdb7c21ab
Status: Downloaded newer image
```

for openjdk:11

```
---> 47a932d998b7
```

Step 2/4 : MAINTAINER osfy

```
---> Running in 782da3d9a384
```

Removing intermediate container 782da3d9a384

```
---> 8f17dd1a4292
```

Step 3/4 : COPY target/spring-petclinic-2.7.3.jar app.jar

```
---> e3acf3fc14f7
```

Step 4/4 : ENTRYPOINT ["java","-jar","/app.jar"]

```
---> Running in df55cf2cbbca
```

Removing intermediate container df55cf2cbbca

```
---> 28440427e82c
```

Successfully built 28440427e82c

Successfully tagged petclinic:latest

Now verify that the *ptclinic* image is available while you execute the *docker images* command:

```
osfy@ubuntu-22-04-1-lts:~/Desktop/spring-petclinic$ sudo
docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
petclinic latest 28440427e82c About a minute ago 707MB
nginx latest 88736fe82739 2 weeks ago 142MB
gcr.io/k8s-minikube/kicbase v0.0.35 7fb60d0ea30e 8
weeks ago 1.12GB
openjdk 11 47a932d998b7 4
months ago 654MB
```

Execute *docker run* command to create a container from the PetClinic image.

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ sudo docker run -d -p
8888:8080 petclinic:latest
b3915b15b7ea8fa94288ea132ec548dee0
488bed412ebe541c39ad13291ce157
```

```
osfy@ubuntu-22-04-1-lts:~/Desktop/spring-petclinic$ sudo docker run petclinic:latest

:: Built with Spring Boot :: 2.7.3

2022-11-30 19:52:05.623 INFO 1 --- [main] o.s.s.petclinic.PetClinicApplication : Starting PetClinicApplication v2.7.3 using Java 11.0.16 on 954ff9a3d0bb with PID 1 (/a
pp.jar started by root in /)
2022-11-30 19:52:05.626 INFO 1 --- [main] o.s.s.petclinic.PetClinicApplication : No active profile set, falling back to 1 default profile: "default"
2022-11-30 19:52:06.308 INFO 1 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.
2022-11-30 19:52:06.339 INFO 1 --- [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 24 ms. Found 2 JPA repository interfaces.
2022-11-30 19:52:06.735 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (http)
2022-11-30 19:52:06.742 INFO 1 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2022-11-30 19:52:06.742 INFO 1 --- [main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9.0.65]
2022-11-30 19:52:06.791 INFO 1 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicationContext
2022-11-30 19:52:06.791 INFO 1 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 1127 ms
2022-11-30 19:52:07.076 INFO 1 --- [main] org.ehcache.core.EhcacheManager : Cache 'vets' created in EhcacheManager.
2022-11-30 19:52:07.083 INFO 1 --- [main] org.ehcache.jsr107.Eh107CacheManager : Registering Ehcache MBean javax.cache:type=CacheStatistics,CacheManager=urn.X-ehcache.
jsr107-default-config,Cache=vets
2022-11-30 19:52:07.097 INFO 1 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2022-11-30 19:52:07.256 INFO 1 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2022-11-30 19:52:07.346 INFO 1 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]
2022-11-30 19:52:07.379 INFO 1 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.6.10.Final
2022-11-30 19:52:07.472 INFO 1 --- [main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}
2022-11-30 19:52:07.547 INFO 1 --- [main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.H2Dialect
2022-11-30 19:52:08.010 INFO 1 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000499: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.pla
tform.internal.NoJtaPlatform]
2022-11-30 19:52:08.016 INFO 1 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2022-11-30 19:52:08.759 INFO 1 --- [main] o.s.b.a.e.web.EndpointLinksResolver : Exposing 13 endpoint(s) beneath base path '/actuator'
2022-11-30 19:52:08.792 INFO 1 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2022-11-30 19:52:08.801 INFO 1 --- [main] o.s.s.petclinic.PetClinicApplication : Started PetClinicApplication in 3.451 seconds (JVM running for 3.806)
```

Figure 4: Container logs

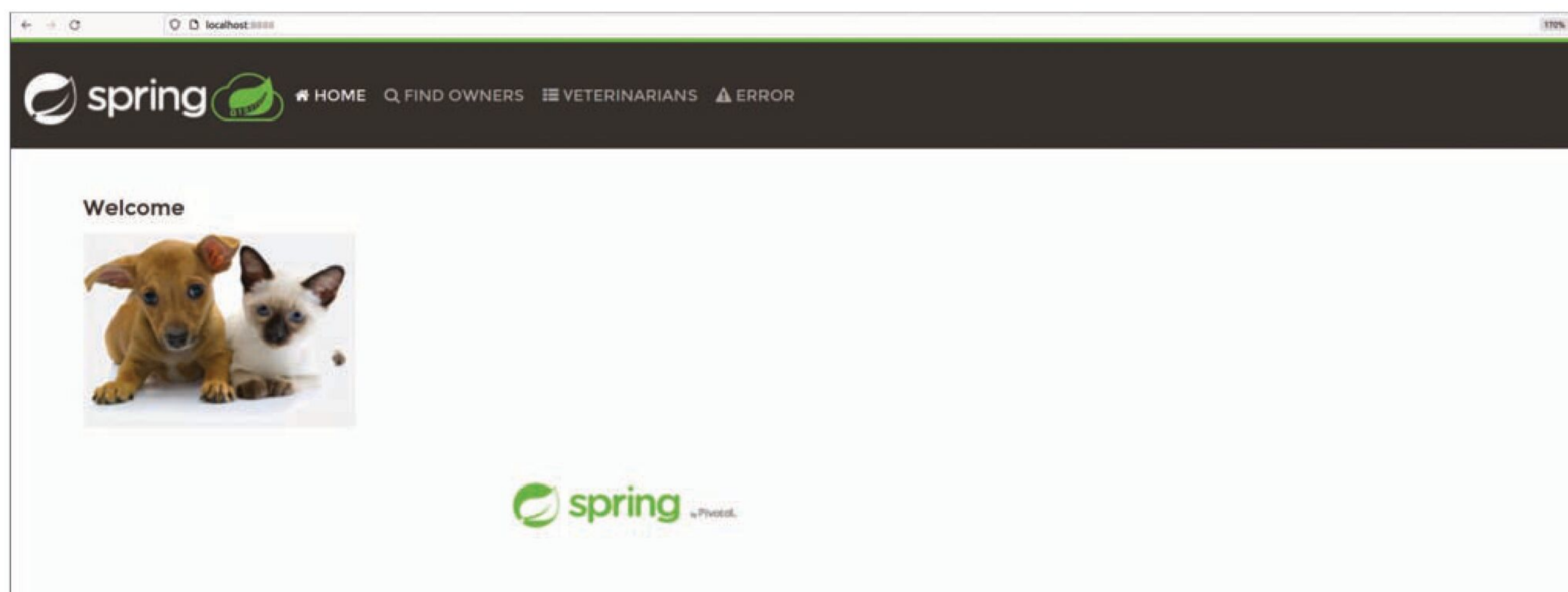


Figure 5: PetClinic home page

Check the available container and its status using `docker ps -a` command:

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ sudo docker ps -a
```

You can see the logs of the Docker container, as shown in Figure 4.

Visit the `localhost:8888` in the browser and you can see that our sample application is ready (Figure 5).

Click on the other *Menu* item available on the page to verify application data.

That's it! We have completed a sample application deployment in a Docker container successfully. **END** 🐧

References

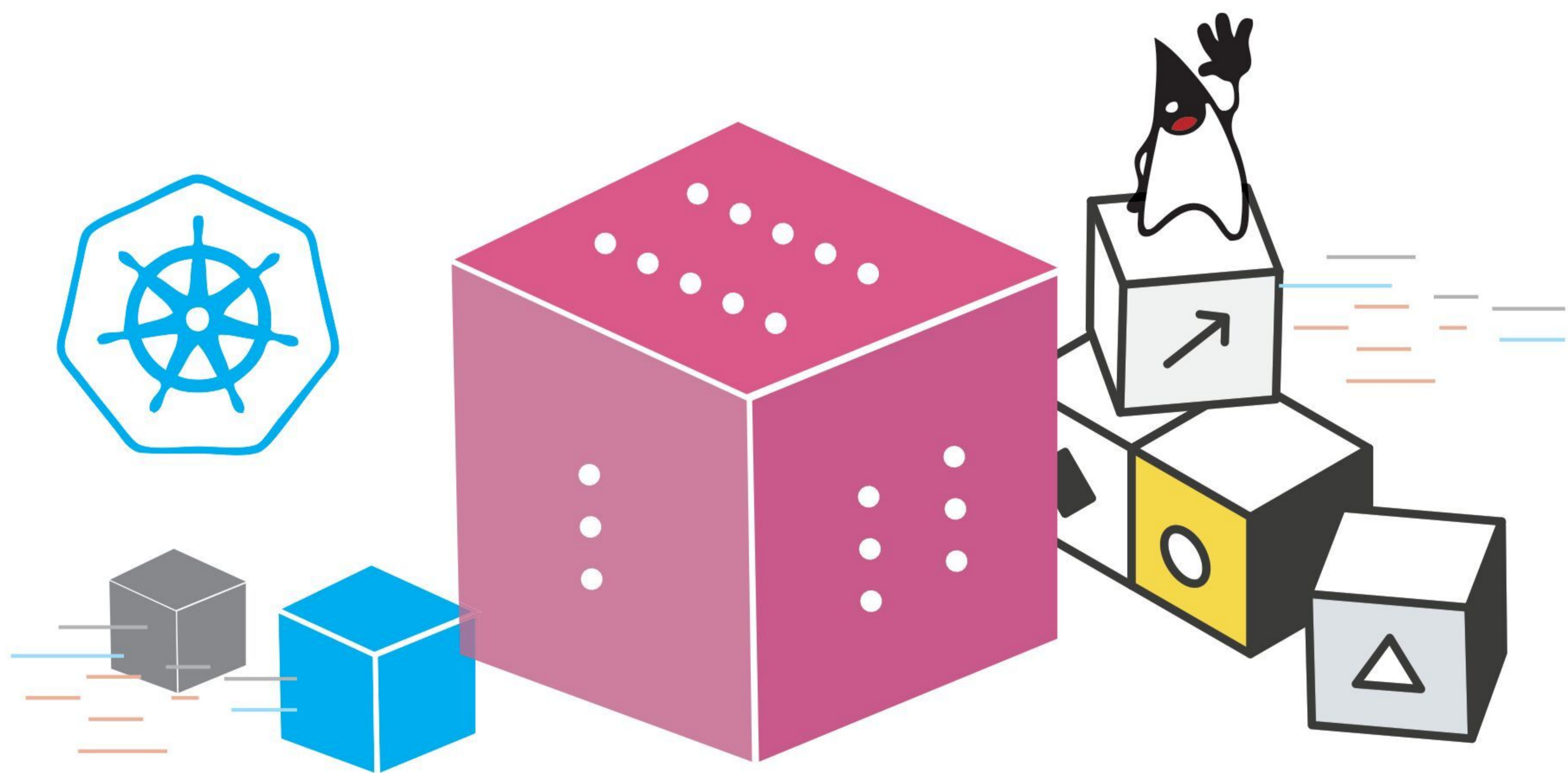
- [1] Docker: Accelerated, Containerized Application Development, <https://www.docker.com>
- [2] Docker documentation, <https://docs.docker.com>
- [3] A sample Spring-based application, <https://github.com/spring-projects/spring-petclinic>

By: Mitesh Soni

The author has written the books 'Hands-on Azure DevOps, Agile, DevOps and Cloud Computing with Microsoft Azure', 'Hands-on Pipeline as Code with Jenkins', and 'Hands-on Pipeline as YAML with Jenkins'.

Deploying a Java App in Kubernetes

This article will tell you how to install minikube and use it for deploying a sample application. It will also tell you how to build a Docker image and run it.



Kubernetes or k8s is an open source, extensible and portable container orchestration platform developed by Google that offers declarative configuration and automation for deployment. It has become one of the most popular tools for DevOps practices because of its capabilities and features for deployment, management, and scaling applications in Kubernetes clusters. We can create a Kubernetes cluster in an on-premise environment as well as on private and public clouds. The public cloud provides managed services for Kubernetes. Google Cloud Platform has Google Kubernetes Engine, AWS has Amazon Elastic Kubernetes Services (EKS) and Azure has Azure Kubernetes Service (AKS) for managing Kubernetes clusters in a cloud environment.

In this article, we will use minikube for sample application deployment. minikube is a tool that is used to run Kubernetes locally. It runs a single-node Kubernetes cluster so that hands-on work can be done on Kubernetes for daily software development.

Some of the important commands that we are going to use in this article are listed below.

Description	Command
Get Kubectl version	<code>kubectl version</code>
Get details on Kubernetes cluster	<code>kubectl cluster-info</code>
Get details about a node	<code>kubectl describe nodeName</code>
Get details about a pod	<code>kubectl describe podName</code>
List of all pods	<code>kubectl get pods</code>
List of all services	<code>kubectl get services</code>
List of all deployments	<code>kubectl get deployments</code>

You will need the following to install minikube on the Ubuntu operating system.

Container or virtual machine manager	Architectures
<ul style="list-style-type: none"> Docker, QEMU, Hyperkit, Hyper-V, KVM, Parallels, Podman, VirtualBox, or VMware Fusion/Workstation 	<ul style="list-style-type: none"> 2 CPUs or more 2GB of free memory 20GB of free disk space

Go to *Settings* and click on the *About* tab to verify the Ubuntu version.

Next we will install minikube on the Ubuntu operating system.

Minikube installation on Ubuntu 22.04.1 LTS

To install the latest minikube stable release on Ubuntu 22.04.1 LTS execute the following commands:

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ curl -LO https://storage.googleapis.com/minikube/releases/latest/minikube_latest_amd64.deb
```

```
% Total    % Received % Xferd Average Speed   Time
Time      Time     Current
```

```
      Dload  Upload  Total  Spent  Left  Speed
100 26.1M  100 26.1M    0    0 1430k    0  0:00:18
0:00:18 --:--:-- 1231k
```

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ sudo dpkg -i minikube_latest_amd64.deb
```

```
Selecting previously unselected package minikube.
(Reading database ... 204624 files and directories currently installed.)
```

```
Preparing to unpack minikube_latest_amd64.deb ...
Unpacking minikube (1.28.0-0) ...
Setting up minikube (1.28.0-0) ...
```

Start your minikube cluster. If you face any issues, stop the cluster, delete it and execute the *minikube start* command again.

Now verify the existing status of minikube by using the *minikube status* command:

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ minikube status
```

We have installed minikube and verified it. The next step is to verify the Java and Maven installation, create a Docker image, and deploy it using Kubernetes YAML in the minikube cluster.

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ minikube stop
👋 Stopping node "minikube" ...
🔴 Powering off "minikube" via SSH ...
🔴 1 node stopped.
osfy@ubuntu-22-04-1-lts:~/Desktop$ minikube delete
🔥 Deleting "minikube" in docker ...
🔥 Deleting container "minikube" ...
🔥 Removing /home/osfy/.minikube/machines/minikube ...
💀 Removed all traces of the "minikube" cluster.
osfy@ubuntu-22-04-1-lts:~/Desktop$ minikube start
😊 minikube v1.28.0 on Ubuntu 22.04
🌟 Automatically selected the docker driver. Other choices: none, ssh
📌 Using Docker driver with root privileges
👍 Starting control plane node minikube in cluster minikube
🚚 Pulling base image ...
🔥 Creating docker container (CPUs=2, Memory=7900MB) ...
🚚 Preparing Kubernetes v1.25.3 on Docker 20.10.20 ...
  ■ Generating certificates and keys ...
  ■ Booting up control plane ...
  ■ Configuring RBAC rules ...
🔍 Verifying Kubernetes components...
  ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟 Enabled addons: storage-provisioner, default-storageclass
🚚 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
osfy@ubuntu-22-04-1-lts:~/Desktop$
```

Figure 1: Start minikube cluster

Java and Maven installation

We need Java and Maven to build a package file for the sample application that we are going to run in a Docker container. Install Java and Maven in the Ubuntu OS. Once installation is completed successfully, verify the Java version using the `java -version` command.

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ java -version
openjdk version "11.0.17" 2022-10-18
OpenJDK Runtime Environment (build 11.0.17+8-post-Ubuntu-1ubuntu222.04)
OpenJDK 64-Bit Server VM (build 11.0.17+8-post-Ubuntu-1ubuntu222.04, mixed mode, sharing)
```

Verify the Maven version using the `mvn -version` command:

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ mvn -version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 11.0.17, vendor: Ubuntu, runtime: /usr/lib/jvm/java-11-openjdk-amd64
Default locale: en_IN, platform encoding: UTF-8
OS name: "linux", version: "5.15.0-53-generic", arch: "amd64", family: "unix"
```

We will use PetClinic as a sample application. PetClinic is a Spring Boot application built using Maven or Gradle. In this article we will build a JAR file and deploy it in a minikube cluster. Visit <https://github.com/spring-projects/spring-petclinic> to know more. Copy the GitHub repository URL from the above URL and use the `git clone` command to clone it locally:

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ git clone https://github.com/spring-projects/spring-petclinic.git
```

Once the source code is available, go to the root of the project using the `change directory` command as given below, and execute the `./mvnw package` command to create a JAR file. At first execution it will try to download multiple dependencies and hence will take some time to complete execution.

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ cd spring-petclinic
osfy@ubuntu-22-04-1-lts:~/Desktop/spring-petclinic$ ./mvnw package
```

```
Warning: JAVA_HOME environment variable is not set.
[INFO] Scanning for projects...
Downloading from spring-snapshots: https://repo.spring.io/snapshot/org/springframework/boot/spring-boot-starter-parent/2.7.3/spring-boot-starter-parent-2.7.3.pom
```

```
Downloaded from central: https://repo.maven.apache.org/maven2/org/springframework/session/spring-session-bom/2021.2.0/spring-session-bom-2021.2.0.pom (3.1 kB at 6.0 kB/s)
[INFO]
[INFO] -----< org.springframework.samples:spring-petclinic >-----
[INFO] Building petclinic 2.7.3
[INFO] -----[ jar ]-----
.
.
[INFO] Building jar: /home/osfy/Desktop/spring-petclinic/target/spring-petclinic-2.7.3.jar
[INFO]
[INFO] --- spring-boot-maven-plugin:2.7.3:repackage (repackage) @ spring-petclinic ---
[INFO] Replacing main artifact with repackaged archive
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 15:36 min
[INFO] Finished at: 2022-12-01T00:56:56+05:30
[INFO] -----
```

Observe the output of Maven command execution where it says `Building jar: /home/osfy/Desktop/spring-petclinic/target/spring-petclinic-2.7.3.jar`.

Now it is confirmed that we have a JAR file ready. The next step is to create a Docker image that contains the JAR file and runtime environment.

So, what is a Dockerfile? Well, it's nothing but the text document that provides build instructions to build the image with your application package.

```
FROM openjdk:11
MAINTAINER osfy
COPY target/spring-petclinic-2.7.3.jar app.jar
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

Place the Dockerfile in the `project root` directory. Now, it is time to build a Docker image.

For testing locally with minikube, we should point the local Docker daemon to the minikube internal Docker registry, and build the image to install it in the minikube cluster: `eval $(minikube docker-env)`.

Building a Docker image and running a container

Execute the `sudo docker image build -t petclinic:latest` command from the `project root` directory.

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ cd spring-petclinic/
osfy@ubuntu-22-04-1-lts:~/Desktop/spring-petclinic$ ls
```

```
build.gradle docker-compose.yml Dockerfile gradle gradlew
gradlew.bat LICENSE.txt mvnw mvnw.cmd pom.xml readme.md
settings.gradle src target
osfy@ubuntu-22-04-1-lts:~/Desktop/spring-petclinic$ docker
image build -t petclinic:latest .
```

The output screen will show you the successful execution of the command.

```
Successfully built 6f78e1ae176e
Successfully tagged petclinic:latest
```

Now verify that the PetClinic image is available while you execute the `docker images` command.

```
osfy@ubuntu-22-04-1-lts:~/Desktop/spring-petclinic$ docker
images
```

REPOSITORY	TAG	IMAGE ID
petclinic	latest	6f78e1ae176e
registry.k8s.io/kube-apiserver	v1.25.3	0346dbd74bcb
registry.k8s.io/kube-scheduler	v1.25.3	6d23ec0e8b87
registry.k8s.io/kube-controller-manager	v1.25.3	603999231275
registry.k8s.io/kube-proxy	v1.25.3	beaaf00edd38
openjdk	11	47a932d998b7
registry.k8s.io/pause	3.8	4873874c08ef
registry.k8s.io/etcd	3.5.4-0	a8a176a5d5d6
registry.k8s.io/coredns/coredns	v1.9.3	5185b96f0bec
k8s.gcr.io/pause	3.6	6270bb605e12
gcr.io/k8s-minikube/storage-provisioner	v5	6e38f40d628d

We will use the Docker images available in the local environment and not from the public or private Docker registry. So set `imagePullPolicy` value to `Never` in the YAML file. Here is the YAML file (for deployment and service) to deploy a sample application in the minikube cluster:

```
apiVersion: apps/v1
kind: Deployment
metadata:
```

```
creationTimestamp: null
labels:
  app: petclinic
name: petclinic
spec:
  replicas: 1
  selector:
    matchLabels:
      app: petclinic
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: petclinic
    spec:
      containers:
        - image: petclinic:latest
          name: petclinic
          resources: {}
          imagePullPolicy: Never
status: {}
---
```

```
apiVersion: v1
kind: Service
metadata:
  name: petclinic
spec:
  type: LoadBalancer
  selector:
    app: petclinic
  ports:
    - protocol: TCP
      port: 8080
      targetPort: 8080
```

Use the `kubectl apply` command to create service and deployment available in the YAML file.

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ kubectl apply -f
petclinic-deployment.yaml
deployment.apps/petclinic created
service/petclinic created
```

The minikube tunnel runs as a process. It helps to expose the Load Balancer Service type. We need to run the `minikube tunnel` command in a separate terminal so the load balancer continues to run until we press CTRL + C. It helps to create a network route on the host. Open a new terminal window or a tab and execute the `minikube tunnel` command:

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ minikube tunnel
```

```
[sudo] password for osfy:
Status:
  machine: minikube
  pid: 21648
  route: 10.96.0.0/12 -> 192.168.49.2
  minikube: Running
  services: [petclinic]
errors:
  minikube: no errors
  router: no errors
  loadbalancer emulator: no errors
Status:
  machine: minikube
  pid: 21648
  route: 10.96.0.0/12 -> 192.168.49.2
  minikube: Running
  services: []
```

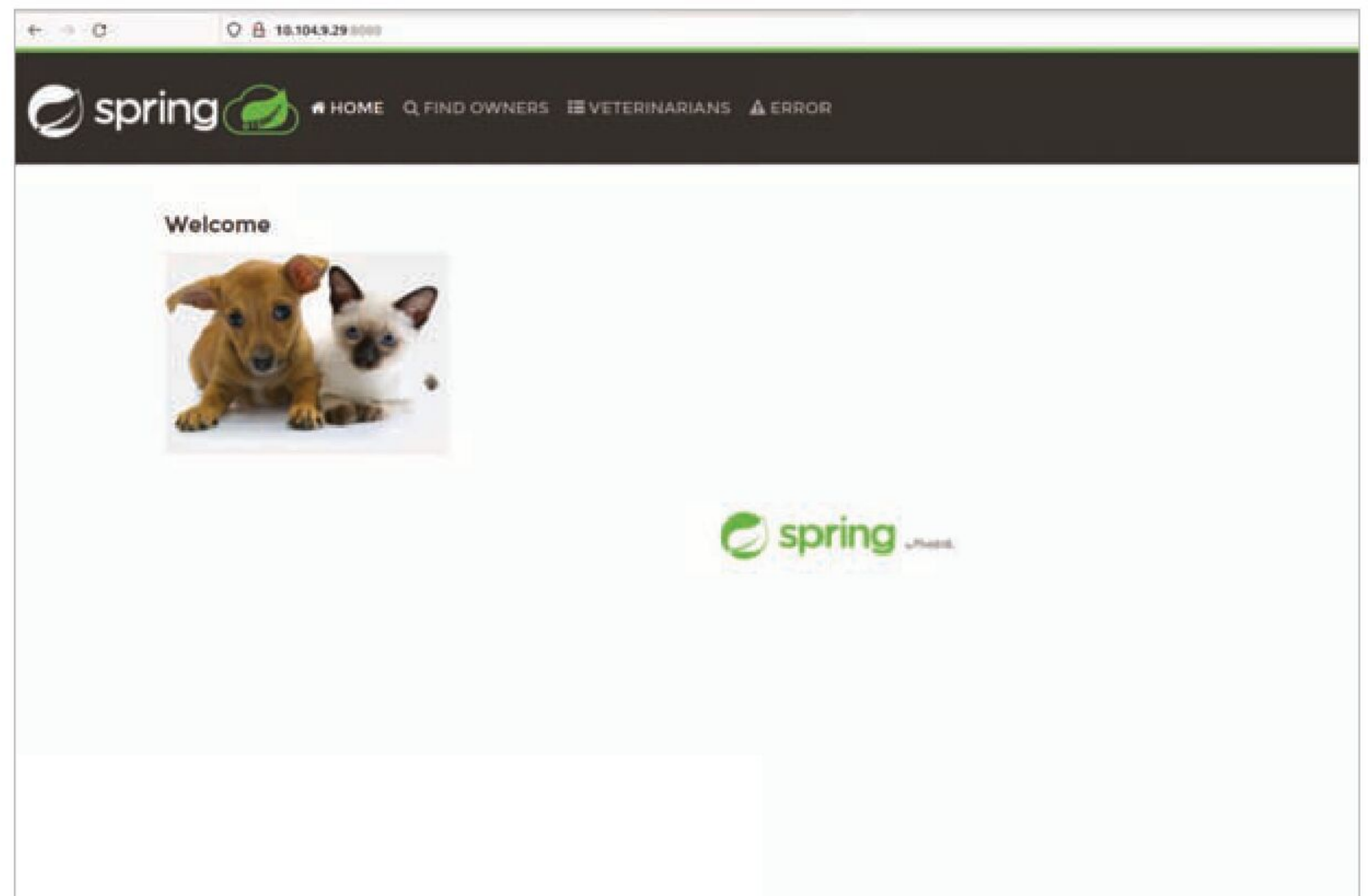


Figure 2: PetClinic home page

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ minikube dashboard
Enabling dashboard ...
  ■ Using image docker.io/kubernetesui/dashboard:v2.7.0
  ■ Using image docker.io/kubernetesui/metrics-scrafer:v1.0.8
Some dashboard features require the metrics-server addon. To enable all features please run:
  minikube addons enable metrics-server

Verifying dashboard health ...
Launching proxy ...
Verifying proxy health ...

Opening http://127.0.0.1:35329/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser..
```

Figure 3: minikube dashboard

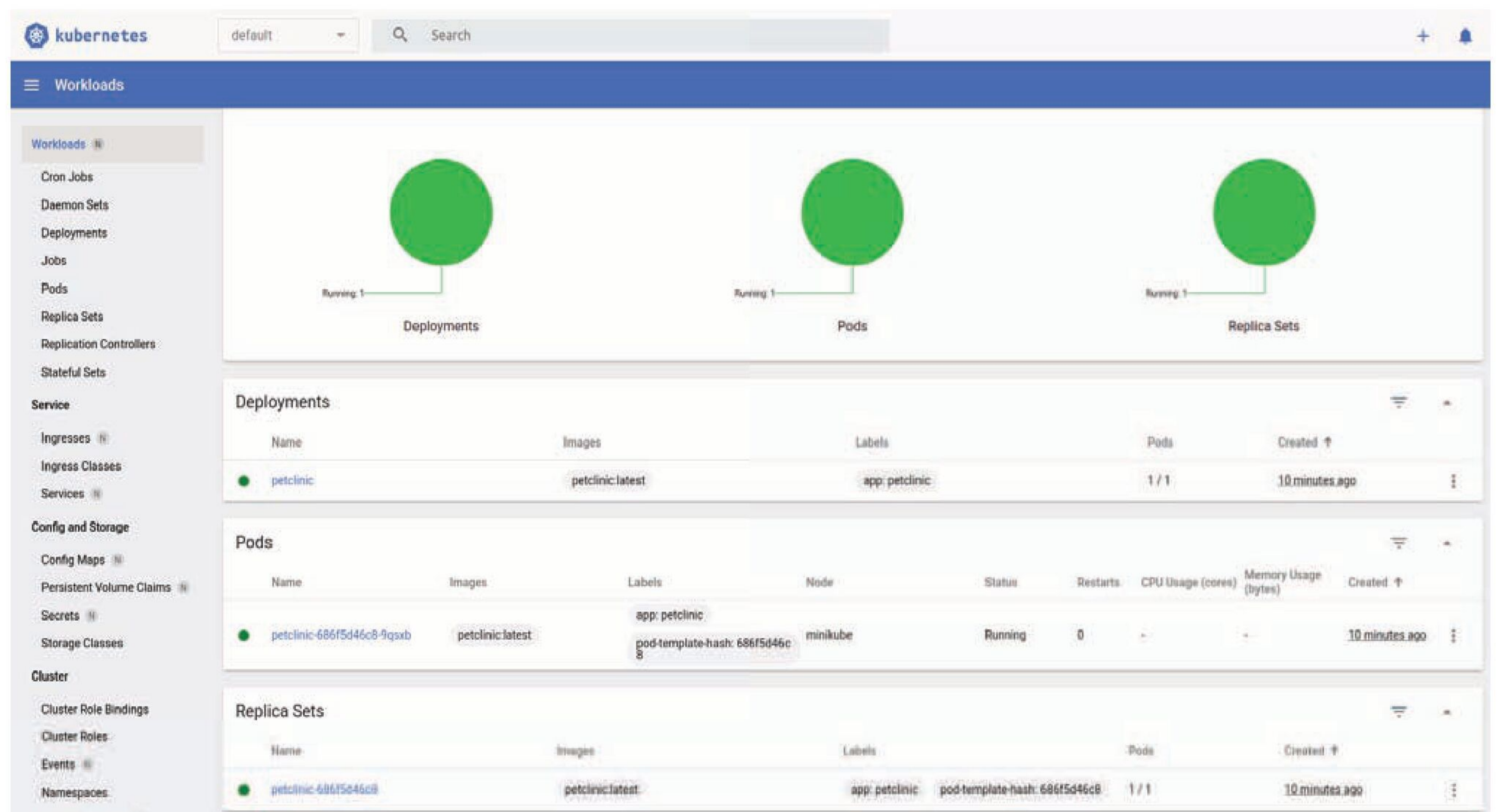


Figure 4: minikube workloads

```
errors:
  minikube: no errors
  router: no errors
  loadbalancer emulator: no errors
```

```
osfy@ubuntu-22-04-1-lts:~/Desktop$ kubectl get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP
kubernetes	ClusterIP	10.96.0.1	<none>
	TCP		443/13h

Let's verify services first:

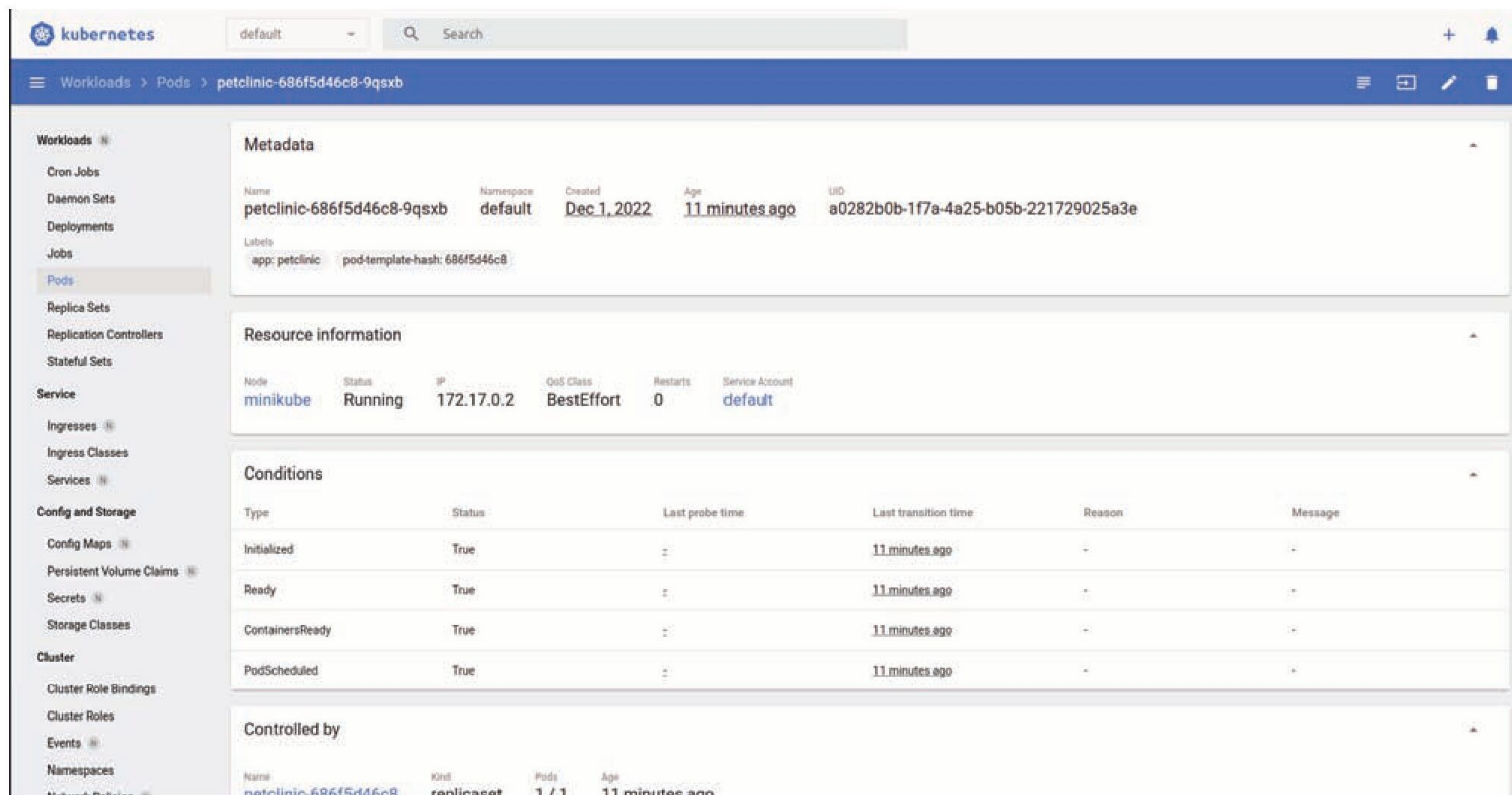


Figure 5: *PetClinic Pod* in minikube dashboard

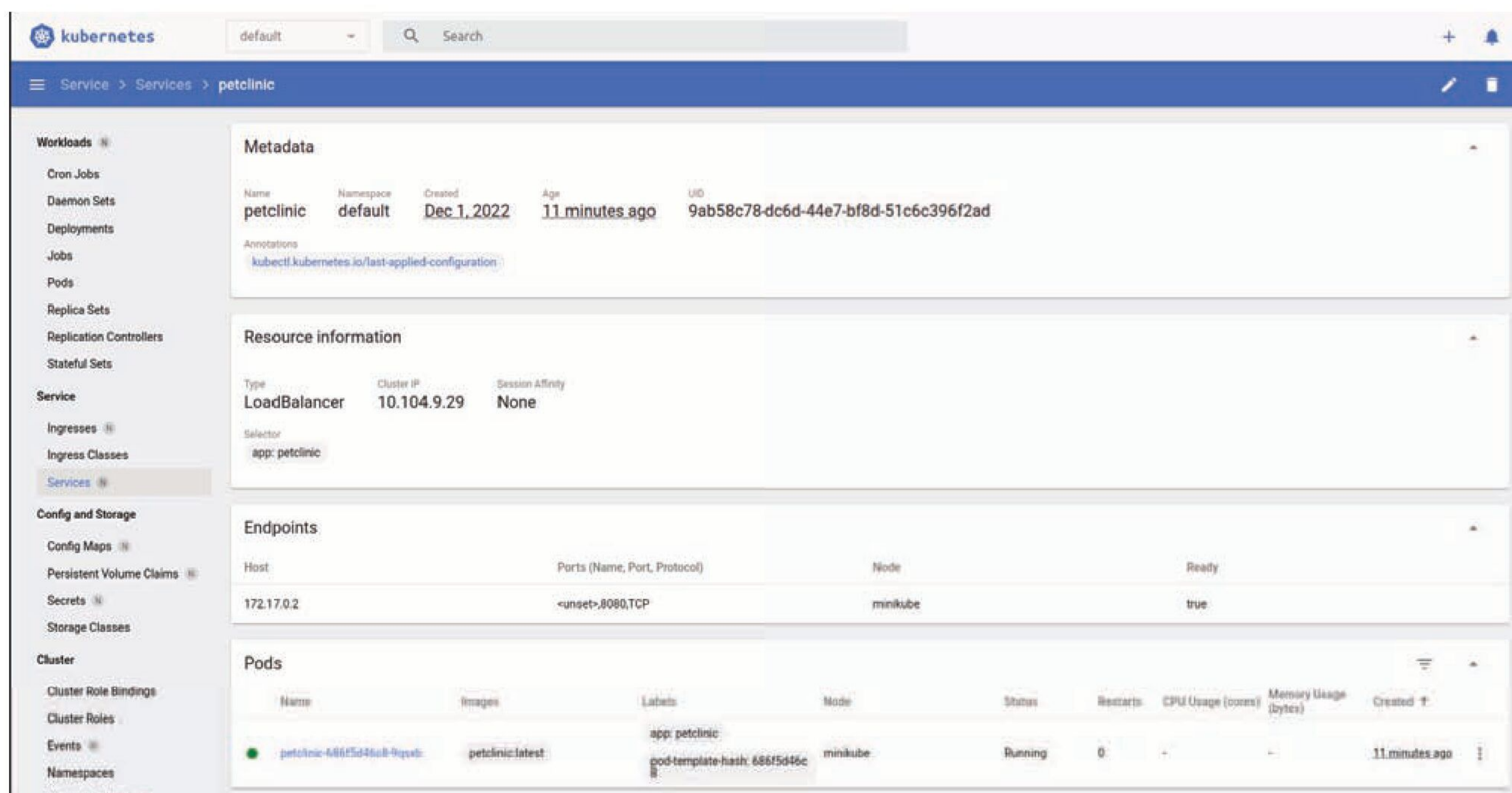


Figure 6: *PetClinic Service* in minikube dashboard

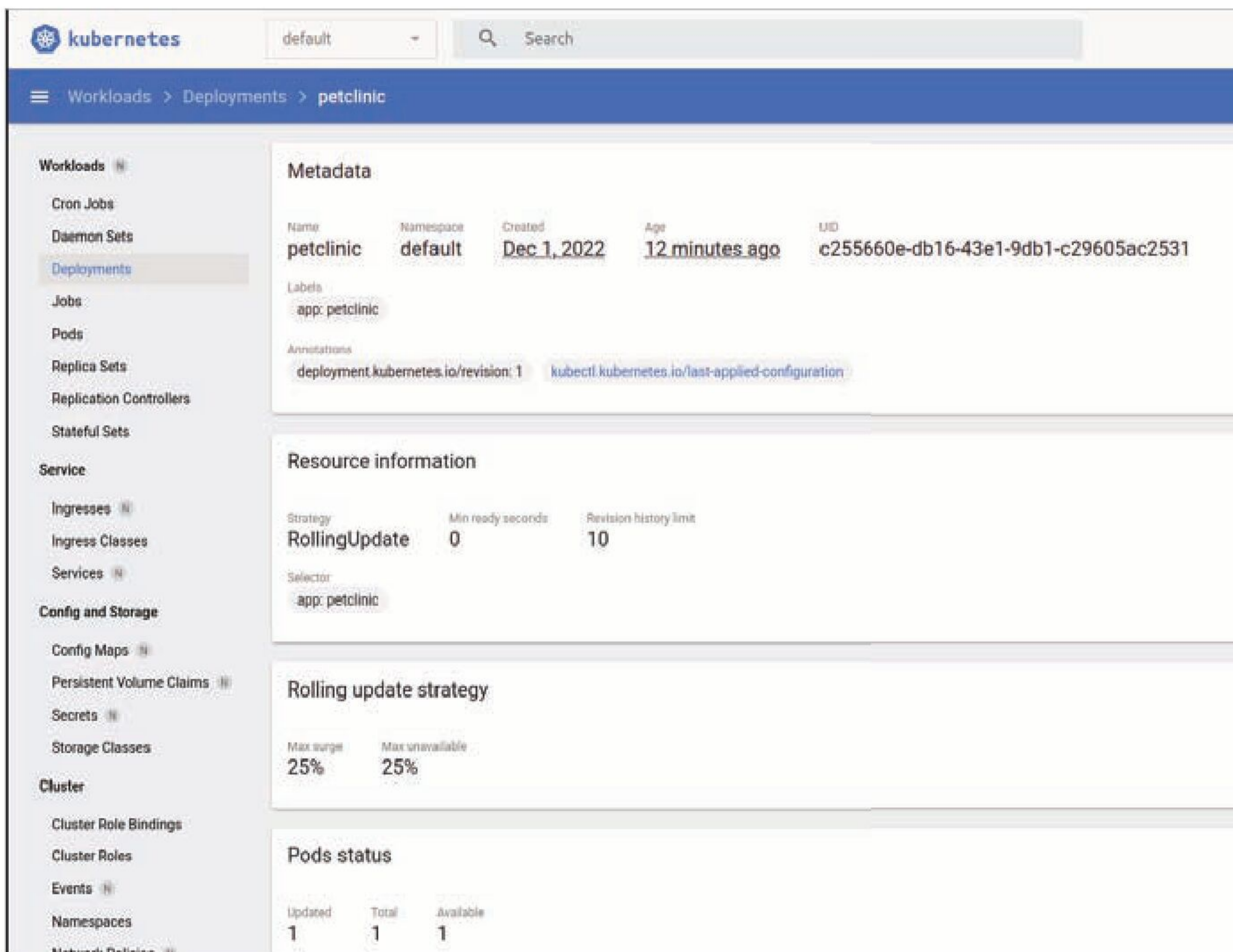


Figure 7: PetClinic Deployment in minikube dashboard

```
petclinic LoadBalancer 10.104.9.29
10.104.9.29 8080:30768/TCP 118s
```

Now let's verify deployments:

```
osfy@ubuntu-22-04-1-lts:~/Desktop$
kubect1 get deployments
```

```
NAME          READY   UP-TO-DATE
AVAILABLE     AGE
petclinic     1/1     1           1
5m3s
```

Next, let's verify pods:

```
osfy@ubuntu-22-04-1-lts:~/Desktop$
kubect1 get pods
```

```
NAME          READY
STATUS        RESTARTS  AGE
petclinic-686f5d46c8-9qsxb 1/1
Running      0          4m56s
```

Visit the localhost:8080 in the browser and our sample application is ready (Figure 2).

Figure 3 shows the verification of the minikube dashboard.

Refer to Figure 4 for verification of workloads with respect to deployments, pods, and replica sets.

Click on *Pods* and go to the *PetClinic Pod* to verify details about it (Figure 5).

Verify the *PetClinic Service* in the *Services* section (refer Figure 6).

Verify the *PetClinic Deployment* in the *Deployments* section. Check the pod status in the same section as seen in Figure 7.

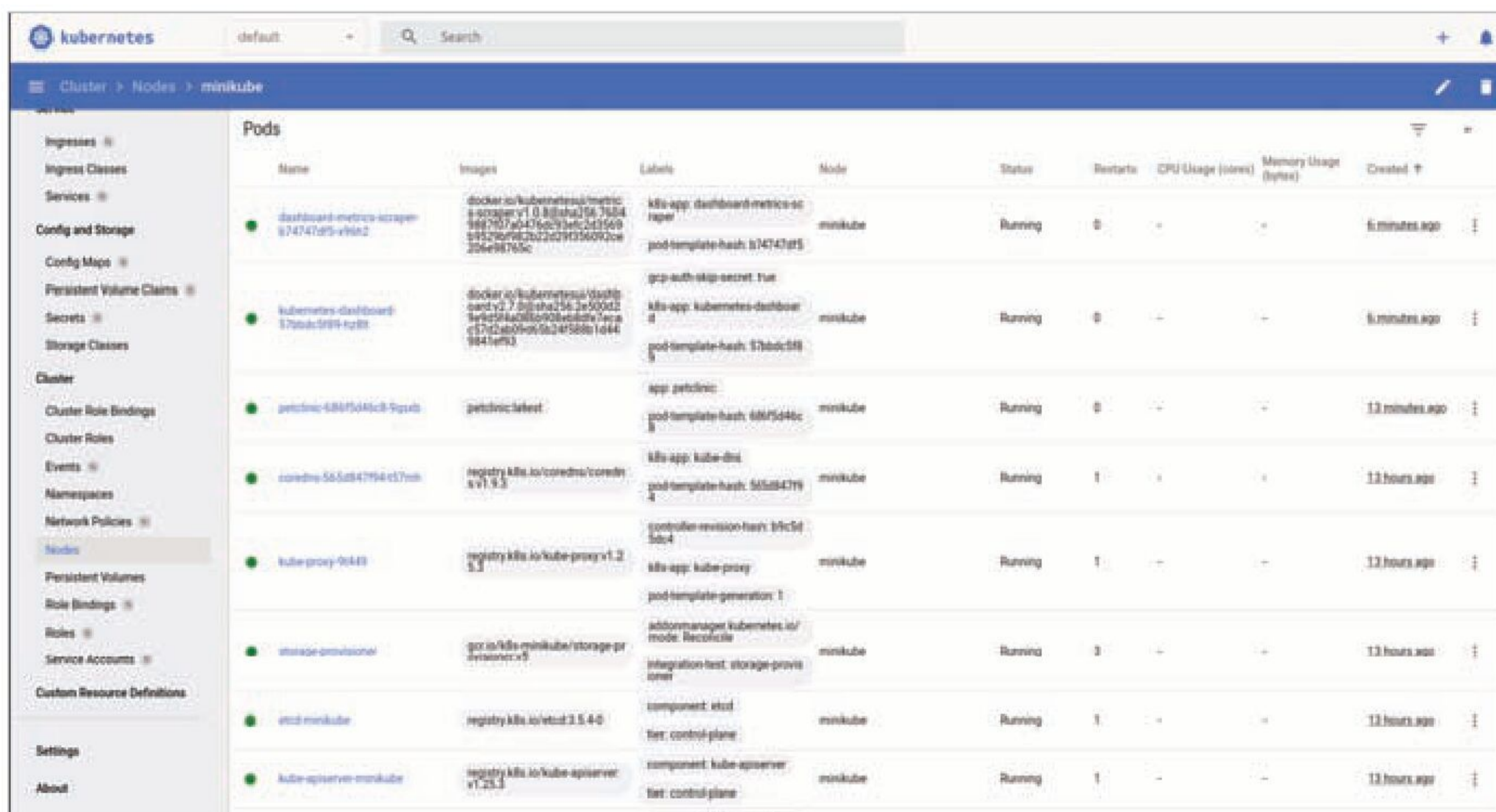


Figure 8: PetClinic Pod in Nodes section

The *Nodes* section gives you details on CPU and memory consumption. It also gives you details on all the pods deployed. Check the *PetClinic Pod* in the same list.

With this, we have completed a sample application deployment in minikube successfully. Hope you enjoyed the learning in this simple 'how to' article. **END** 🐧

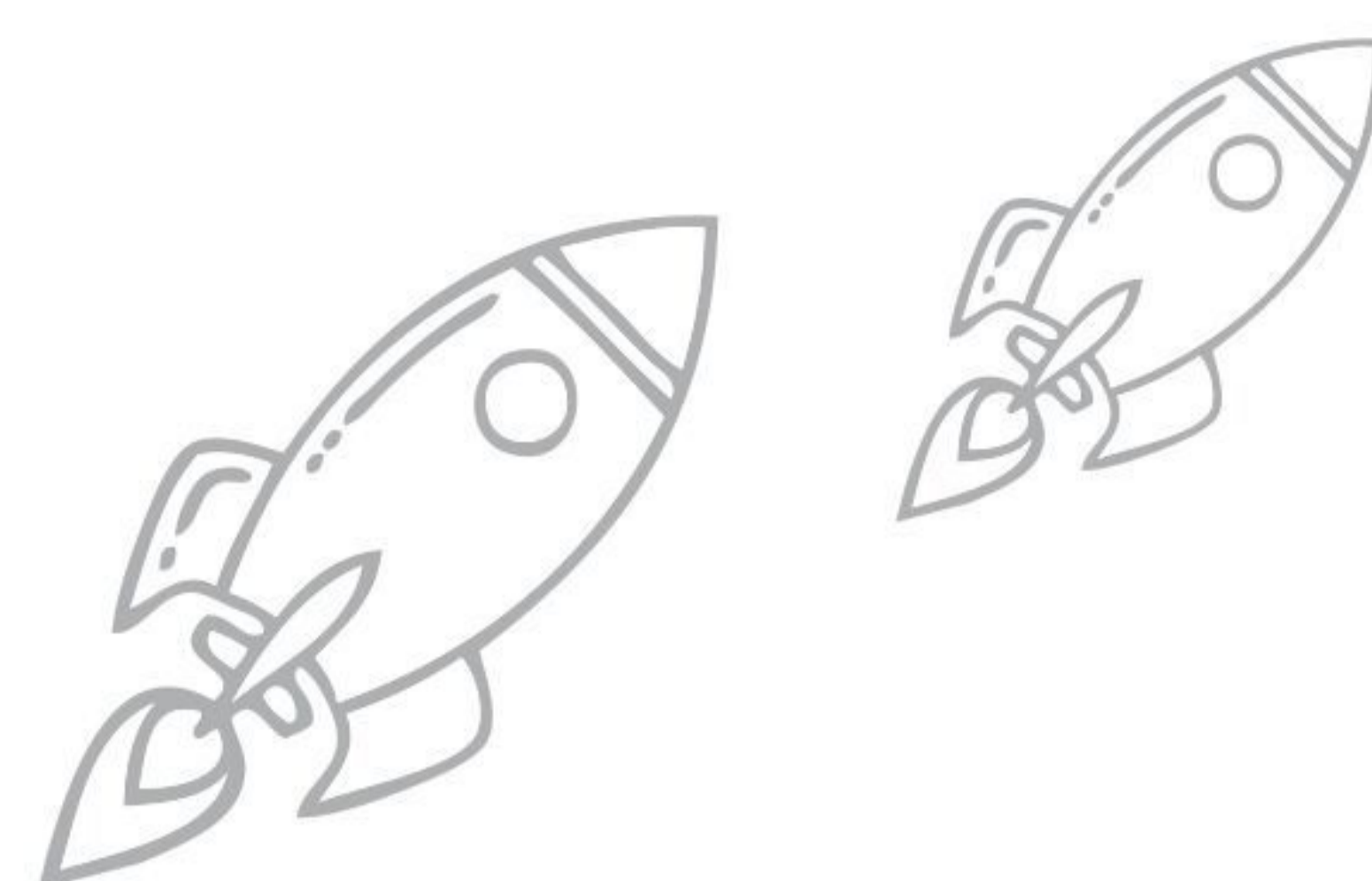
References

- [1] Minikube documentation, <https://minikube.sigs.k8s.io/docs/start/>
- [2] Docker: Accelerated, Containerized Application Development, <https://www.docker.com>
- [3] Docker documentation, <https://docs.docker.com>
- [4] A sample Spring-based application, <https://github.com/spring-projects/spring-petclinic>
- [5] Deployments, <https://kubernetes.io/docs/concepts/workloads/controllers/deployment/>
- [6] Service, <https://kubernetes.io/docs/concepts/services-networking/service/>

By: Mitesh Soni

The author has written the books 'Hands-on Azure DevOps, Agile, DevOps and Cloud Computing with Microsoft Azure', 'Hands-on Pipeline as Code with Jenkins', and 'Hands-on Pipeline as YAML with Jenkins'.

Getting Started with Kubernetes: The Ultimate Guide



Learn how to get started with Kubernetes with the help of this simple tutorial.



An orchestrator is a piece of software that distributes container workloads over several servers or nodes. Its primary function is to orchestrate container workloads. To prevent any kind of downtime from occurring, it is necessary to ensure that the containers have access to all the appropriate resources, that the correct number

of containers is always on hand, and that additional steps, such as rolling deployments, are done. Kubernetes is the container orchestrator that has had the highest amount of adoption recently.

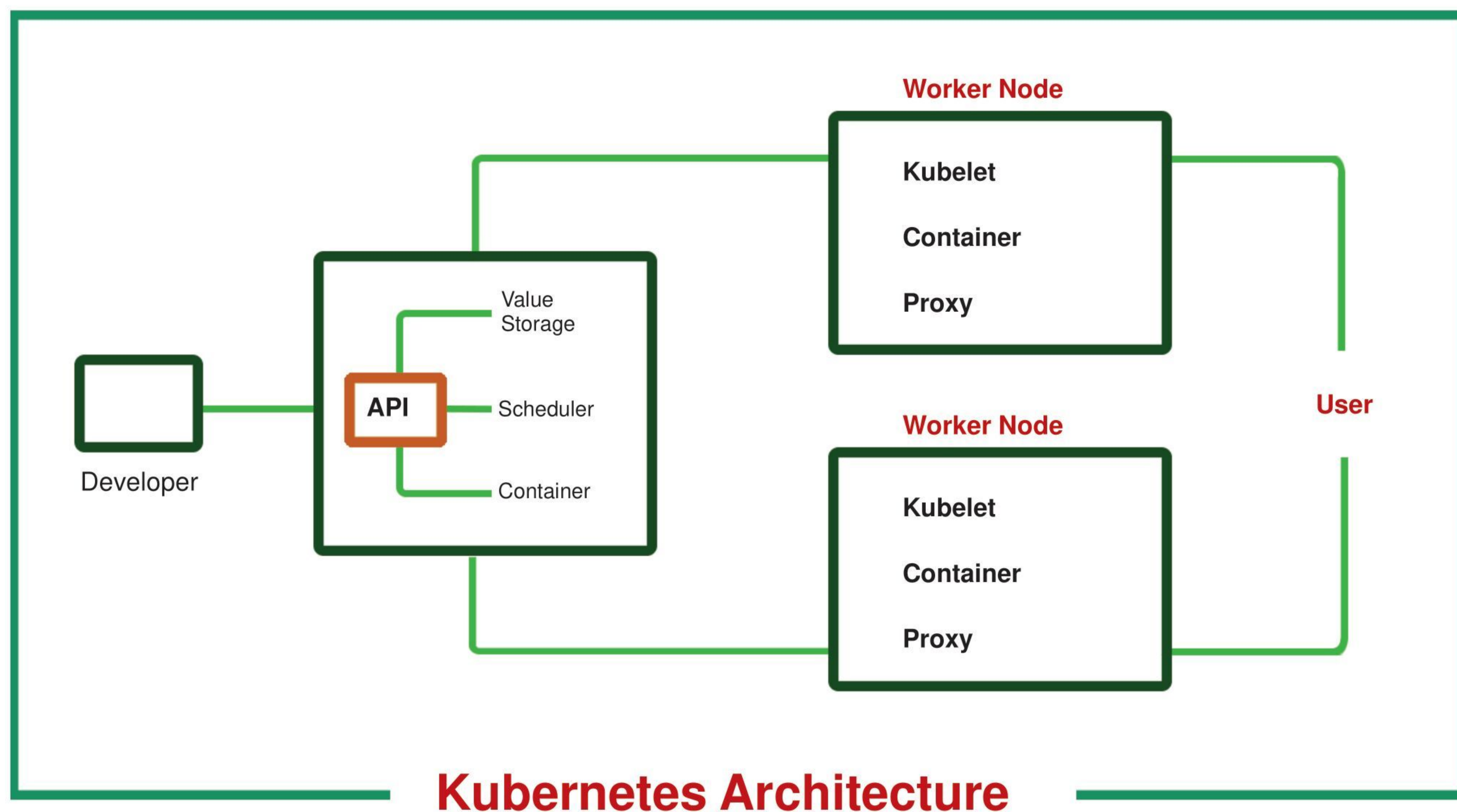
Before moving on to Kubernetes, let's get a grasp on containers and see how they work. Containerization refers to the process of containing the source code of an application as well as any

dependencies it has into a single image, and then running that image within a computer environment that is isolated from any other environments.

It is common practice to credit Docker as the breakthrough that led to the development of containerization. This is because Docker made it possible to package software into a portable virtual container. Containerization provides the portability of code to cloud services such as Google Cloud Platform, Amazon Web Services, and Digital Ocean.

However, complications arise as the number of containers increases. Deploying and managing hundreds of containers across several servers or virtual machine instances can be a very time-consuming process. That's what led to the development of container orchestration tools like Kubernetes.

Kubernetes, which is often referred to as K8s, was first released by Google in 2015, and is now being maintained by an open source community (of which Google is also a member). Kubernetes is a system that, by using a combination of application programming interfaces and command line tools, enables the coordination of container processes that span several hosts. The process of deploying, scaling, and maintaining container clusters across several hosts is simplified as a result of using this tool. Because it handles problems like configuration management,



Kubernetes Architecture

Figure 1: Kubernetes setup

service discovery and job scheduling, which occur with the deployment of microservices, it is often utilised as a host for such endeavours.

In addition to the original Kubernetes codebase, there are a number of cloud providers and distributions that provide their very own versions of the Kubernetes container orchestration system.

Here are a few Kubernetes attributes and qualities:

- Performs a number of hard tasks automatically, in addition to regulating server hosting and startup.
- Services like data storage, networking security, and management are controlled by containers.
- Maintains a constant vigil on the states of the nodes and containers under its care.
- Rollbacks are handled automatically for any modifications that are unsuccessful.
- Puts together an application mount for storage and adds a storage system to utilise with it.

Why is Kubernetes considered imperative?

Kubernetes has seen increased demand as a result of the trend towards ‘microservices’, which are defined as applications that are smaller and more agile. Containers are the appropriate host for these decoupled microservices, since they are able to manage dependencies, operate independently, are not dependent on any particular platform, and may be destroyed when they are no longer required.

It is not unheard of for an application to be composed of hundreds, or even thousands, of microservices. This kind of program often has a great number of moving pieces. If we manage all of these varied components using custom-written programs or scripts, then increasing the size of these microservices while maintaining availability will be a challenge. Because of this, there is a need for a better technique of component management.

A glance behind the scenes at how everything works

When a K8s cluster is in operation, one of the nodes is designated as the master, and it is this node that is responsible for making all the crucial decisions affecting the cluster. The other nodes, which are referred to as slaves or worker nodes, are responsible for carrying out the workloads for the applications in accordance with the master’s directives.

A straightforward Kubernetes design

Figure 1 presents an illustration of a fundamental K8s setup, which consists of a single master node and two worker nodes.

Master node(s)

The master node acts as the cluster’s primary coordinator. It is responsible for making decisions on the general status of the cluster as well as the duties that are delegated to each of the worker nodes. In order to make a master node operational and ready for usage, the following four operations need to be running on it.

1. API server

- User queries received through `kubectl` are routed via this node, making it the cluster's primary gateway.
- It controls who can access resources and make requests by acting as a gatekeeper for authenticating and verifying requests.

2. Scheduler

- Determines on which node the next pod will be spun up. However, this action does NOT spin up the pod itself; Kubelet is responsible for this step.

3. Controller manager

- Identifies when the cluster's status has changed (for example, when pods have died) and attempts to return it to its initial, created state.
- In case the launch of a pod is unsuccessful for whatever reason, the Controller Manager will request that the Scheduler choose a node for the launch of a new pod. After that, Kubelet will begin the spinning of the new pod.

4. etcd

- This is the key-value storage used by the cluster.
- It is the location where any adjustments that have been made to the cluster are recorded.
- This location is used only for the storage of information pertaining to the cluster itself; no application data is ever stored there. It is important to remember that the master node is not a worker but rather the command and control hub for the cluster. The status information is saved in etcd so that the previously described procedures may acquire knowledge of the cluster.

Slave/worker node(s)

In order for Kubernetes to be able to connect with the nodes and launch pods

on its own, the Central Concentration of Servants and Employees (s) component of Kubernetes mandates that three different node processes be installed on each and every worker node. The following three steps are required.

1. Kubelet a.k.a. *kubelet*

- Facilitates two-way communication between the node and the container.
- It is responsible for reading configuration files and starting the pod by using the container runtime that is provided by the node 2 server (more on this later!). It's the OS for containers.

2. Container runtime

- A computer setup that already has a container runtime installed (e.g., Docker, containerd)
- Kube's proxy, often known as kube-proxy, comes in third.

3. Kube Proxy a.k.a. *Kube-proxy*

- It is located between computers and intelligently forwards requests, either inside a node or to other nodes.
- Since we now have an understanding of how K8s operates, we are able to investigate the Kubernetes building pieces that are most often seen throughout the process of application deployment.

Components of Kubernetes

Now that we understand how K8s functions, we can examine the Kubernetes building blocks most often seen during application deployment.

1. Pod

- The smallest building element that makes up a Pod K8, and the location where a real instance of your software is deployed.
- Over the top of a receptacle is an abstraction.
- Every pod has its own one-of-a-kind IP address (public or private).
- Until the pod service is rebuilt, the IP address will be changed.

2. Service

- Despite the ephemeral nature of pods, the service provides a means to 'give' them a permanent IP address.
- Even if the pod is destroyed and then re-created while it is still connected to the service, it will keep the same IP address.
- It redirects traffic to pods while maintaining a single IP address, functioning in a manner that is similar to that of a load balancer.
- Similar to load balancers, the service may either be internal or external. An external service is one that is exposed to the general public (has public IP), while an internal service is one that is intended for use with internal applications (has private IP).

3. Ingress

- Because of services, we may now have a web service that is listening on a certain port (for example, 8080) at an IP address (for example, 10.104.35). Unfortunately, it is not possible to browse a public-facing application by using the address `http://10.104.35:8080`. This is because it is not a practical address.
- This indicates that we are in need of a domain-based entry point (such as `https://my-domain-name.com`) that directs traffic to the service (such as `http://10.104.35:8080`).
- In essence, Ingress makes available to internal cluster services both HTTP and HTTPS routes that originate from the outside.
- Due to SSL termination, traffic destined for the service and its pods will not be encrypted (also known as SSL offloading).
- Just so there is no confusion, constructing an Ingress resource serves no use on its own. An Ingress Controller is required in addition to an Ingress in order for the latter to be finished.

What is an Ingress Controller?

- The Ingress Controller is responsible for load balancing and also controls the traffic that enters and exits the cluster.
- This controller is responsible for services that need to connect with other services that are located outside of the cluster.

ConfigMap

- To be more specific, it is a configuration file that may be edited by the user.

Secret

- It is similar to a configuration file, but designed to contain private data. It is Base64-encoded.

Volumes

- It is capable of retaining data forever and offering useful benefits.
- Because pods are ephemeral, it is necessary to store data in volumes so that pods of varied ages may continue to refer to the same state of the application.
- It provides capabilities that are analogous to that of an ‘external hard drive’, but for your pods.
- The volumes of your pods may either be stored locally on the same node or remotely on another node (e.g., cloud storage, NFS).

Deployment

- In most cases, we focus our attention on deployments rather than on pods.
- In case of problems with the main deployment, most also have a copy accessible as a backup.
- Stateful programs, on the other hand, can’t have their components replicated, which means that things like databases can’t be cloned either. We will need the Kubernetes module known as StatefulSet in order to do this. Due to the complexity of this undertaking, databases should almost always

be stored on servers that are geographically distinct from the Kubernetes cluster.

Utilisation of the Kubernetes platform

In order to continue with our introduction to Kubernetes, let’s take up the *New York Times* as a case study.

- Following the publisher’s decision to close its data centres, smaller deployments were carried out, and virtual machines (VMs) were placed in charge of managing the applications.
- They got to work in order to produce brand new instruments. However, at some time they realised that they were missing the target since they treated Amazon like any other data centre.
- The programmers quickly responded to the issue and came up with an elegant solution. The panel discussed the possibility of using Google Cloud Platform to deploy Kubernetes.
- The use of Kubernetes brought about a number of benefits, including the following:
 - A quickening of the pace in performance and delivery.
 - The time required for deployment was reduced from minutes down to seconds.
 - The updates were sent on their own and only when absolutely required.
 - Increased portability as well as an improved and more standardised approach of deployment for all engineers.

In conclusion, the *New York Times* has shifted from a system that was based on tickets to an automated system that is based on Kubernetes in order to request resources and schedule installs.

Kubernetes demo

In order to proceed with the tutorial on getting started with Kubernetes, let’s

get directly into a demonstration.

1. Open the command prompt in Ubuntu.
2. The installation of all requirements may be completed with the following command:

```
$ sudo apt-get update
```

```
$ sudo apt-get install -y apt-transport-HTTPS
```

3. Make use of the following command in order to install the Docker dependency:

```
$ sudo apt install docker.io
```

4. To activate Docker and get it up and running, use the following commands:

```
$ sudo systemctl start
```

```
$ sudo apt-get install curl
```

Obtain the Kubernetes installation key, and then add it to your machine using the following instructions:

```
$ sudo curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add
```

It is possible to change permissions by using the command that is listed here:

```
$ sudo chmod 777 /etc/apt/sources.list.d/
```

After that, create a repository by inserting the following information into the `/etc/apt/sources.list.d/kubernetes.list` file:

```
deb http://apt.kubernetes.io/
Kubernetes-xenial main
```

Put your modifications in a safe place, and then close the file.

To configure Kubernetes, use the following commands:

```
$ apt-get update
$ apt-get install -y kubelet kubeadm
kubect1 Kubernetes-cni
```

5. Before we can initialise the master node, we need to first switch the nodes by executing the following command:

```
$ sudo swapoff -a
```

6. The following command should be used to initiate the master node:

```
$ sudo kubeadm init
```

Following are three sets of instructions. Simply cut and paste, and press 'Enter'.

```
$ mkdir -p $HOME/.kube
$ sudo cp -i /etc/Kubernetes/admin.conf
$HOME/.kube/config
$ sudo chown $(id -u):$(id -g) $HOME/.
kube/config
```

7. To release pods, just follow these command steps:

```
$ $ sudo kubect1 apply -f https://raw.
githubusercontent.com/coreos/flannel/
master/Documentation/kube-flannel.yml
$ sudo kubect1 apply -f https://raw.
githubusercontent.com/coreos/flannel/
master/Documentation/k8s-manifests/
kube-flannel-rbac.yml
```

8. You can view a complete list of dropped pods by typing the following command into your console:

```
$ sudo kubect1 get pods -all-namespaces
```

9. To install an NGINX service (and to make the service available via port 80), the following commands need to be executed:

```
$ sudo kubect1 run --image=nginx Nginx-
app --port=80 --env="DOMAIN=cluster"
```

```
$ sudo kubect1 expose deployment nginx-
app --port=80 --name=nginx-HTTP
```

10. In order to see the services that are available, use the following command:

```
$ sudo docker ps -a 
```

By: Sufyan bin Uzayr

The author, apart from being a writer, is an educator and entrepreneur. He holds multiple degrees in management, information technology and humanities. He has written seven books so far and is currently the director of a multinational IT company. As an educator, he has taught at universities and institutions worldwide.

FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK

We Value Your Feedback

We love to hear from you as **OSFY** consistently strives to make its content **informative, interesting and engaging.**



Please share your feedback/ thoughts/ views via email at osfyedit@efyindia.com.

We welcome your comments/ suggestions and encourage you to send them to:
The Editor, D-87/1, Okhla Industrial Area, Phase 1, New Delhi-110020.

OpenSource
THE COMPLETE MAGAZINE ON OPEN SOURCE **ForYou**

FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK FEEDBACK

Free Tools and Libraries that Help to Keep Docker and Containers Secure



Let's take a quick look at the free tools and libraries that are available for the security of Docker and containers.

Docker is a containerization solution that lets programmers encapsulate a service together with its dependencies, prerequisites, libraries and operating system. It facilitates app deployment and portability by isolating application code from underlying system code.

Over the past few years, more and more apps are being containerized rather than being hosted on virtual machines, thanks to the use of Docker. The latter is essential to this shift because it facilitates the smooth adoption of containerization technologies. Use cases

for Docker have emerged across every conceivable business sector, no matter how big or small.

By decomposing a code into smaller and more manageable pieces that can communicate with one another via REST APIs, a microservices design helps developers overcome development challenges with Docker technology. Each feature may often be implemented as its own service, meaning it can run separately from the others. As a result, businesses can hasten the pace of release cycles, expand operations on demand, and implement code

changes with minimal disruption to service. Among the many applications of Docker, one of the most common is the transition from monolithic to microservices based architectures.

One of the most interesting applications of microservices architecture is the container. A container is a standardised software package that separates an application's runtime from the system on which it runs by enclosing it with all of its dependencies and resources. Containers are different from virtual machines in that they only virtualise software layers above the OS.

Docker's positive impact on enterprises and real world scenarios

Because of its mobility and ecosystem, Docker has grown to be synonymous with containerization. It has been adopted and supported by all the main cloud providers including GCP, AWS, Azure and many others. Organisations are saving a lot of money by using containerization and running several containers on a single host. Since Docker enables service deployment on inexpensive commodity hardware, it may be used to avoid spending money on costly servers.

Docker is prominently used for creating and managing containerized development environments with Docker images, which enable rapid provisioning and consistent performance over the course of a project's life span. In addition, it provides comprehensive assistance for all development-related tools, frameworks, and technologies.

Docker environments also make it easier to use tools like CI/CD, A/B testing, and webhooks. Thus, it is simple to include Bitbucket or GitHub repos with the development environment, allowing for automated builds to be generated from the source code and afterwards pushed to the Docker Repo.

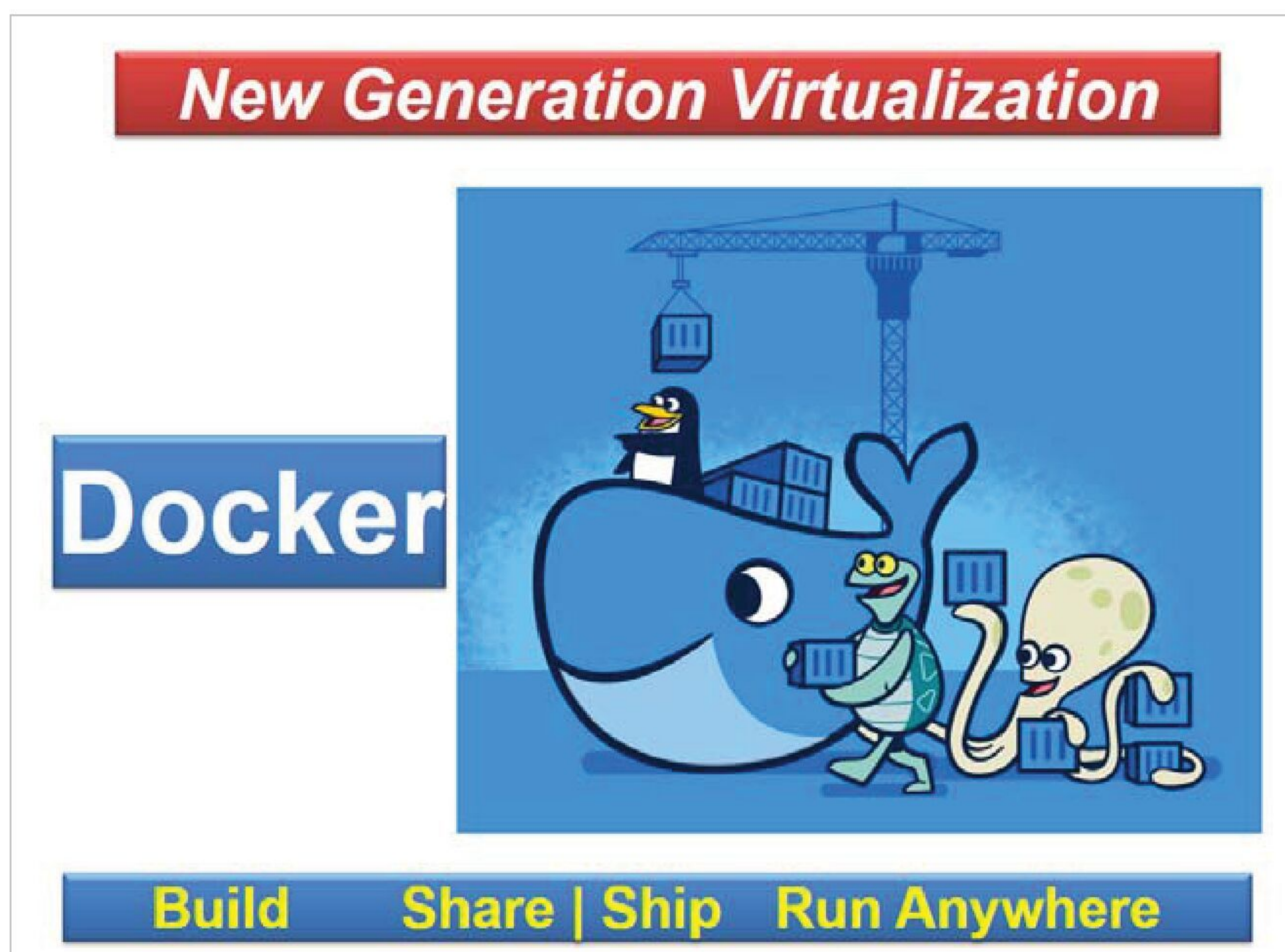


Figure 1: Docker platform

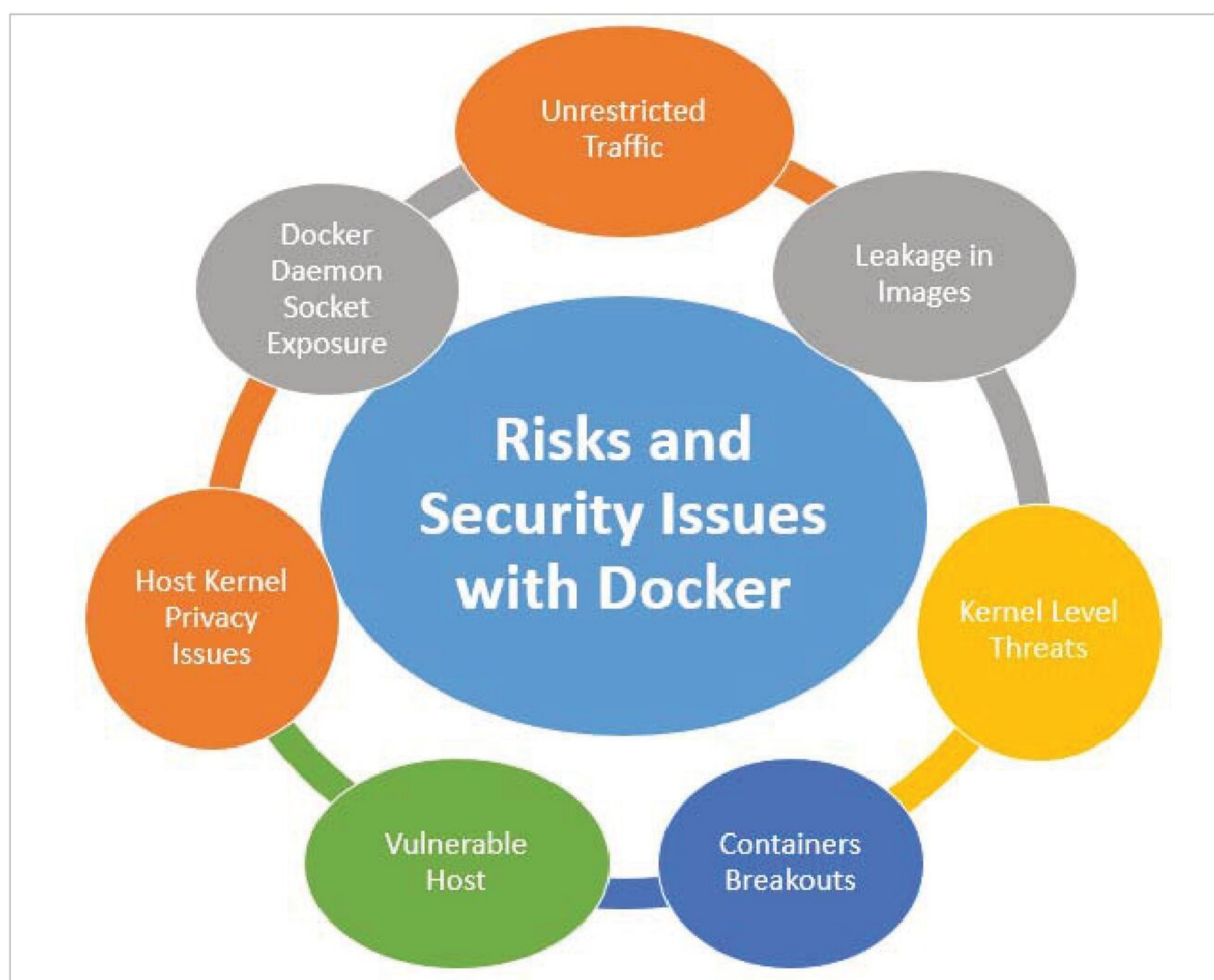


Figure 2: Key risks and security issues with Docker

One of the most important ways to put Docker to use is in the development and deployment of software. There is less risk of unexpected issues arising due to configuration variations between the test environment and the production environment when developers conduct

tests in the same environment where the apps will run in production.

Security and integrity aspects in Docker containers

Once attackers have compromised a host, they frequently have the resources to go laterally and compromise more

Table 1: Key tools and libraries that help with security in a Docker environment

Library / Tool	Usage	URL
Docker Bench for Security	Audit of Docker containers and benchmarking	https://github.com/docker/docker-bench-security
Cilium	Kernel layer security	https://github.com/cilium/cilium
Anchore	Inspection of container security with integration of policies	https://github.com/anchore/anchore-engine
OpenSCAP Workbench	Integration of security and privacy policies	https://www.open-scap.org/
Dagda	Scanning of viruses, malware, Trojans and other malicious traffic in Docker containers	https://github.com/eliasgranderubio/dagda
Notary	Integration of cryptography and deep level security	https://github.com/notaryproject/notary
Grafeas	Internal policies for privacy and security	https://grafeas.io/
Sysdig Falco	Deep level visibility and scan of containers with behavioural activity analysis	https://github.com/falcosecurity/falco
Banyanops Collector	Evaluation of Docker on different perspectives and dimensions	https://github.com/banyanops/collector
Dockscan	Security scanner against vulnerability	https://github.com/kost/dockscan
Batten	Docker audit toolkit	https://github.com/dockersecuritytools/batten
Inspec	Testing, audit and deep level analysis of the Docker environment	https://github.com/inspec/inspec

containers. A container's ability to access the system file directory might weaken the effectiveness of security measures. In rare cases, attackers who get root access to a container can use it to gain root access to the host too.

When it comes to container security, Docker has you covered at every stage: during development, during deployment, and during orchestration. However, the Docker daemon, container runtime, Dockerfile, and base images are all potential entry points for attackers.

So Docker does come with a few risks and vulnerabilities:


- Unsafe and unrestricted traffic with vulnerable communications
- Malicious and vulnerable container images
- Host system security loopholes
- Unrestricted access
- Exposure of Docker daemon socket

- Privacy leakages in Docker images
- Vulnerabilities with host kernel
- Container breakouts
- Kernel level threats

Free tools and libraries for Docker and containers

Though Docker is a secured environment, there is huge scope for research in its privacy, integrity and security aspects. Lots of libraries and frameworks are available and still under development to cope with the security leakages in a Docker environment (see Table 1). The Common Vulnerabilities and Exposures (CVE) database and the National Vulnerability Database (NVD) are

the key databases that receive a lot of attention from auditors and are used as standards by other organisations. After the container image is scanned, its contents are exposed, and comparisons are made against the vulnerability manifests in order to assess its security posture.

Organisations can benefit a lot from automating container audits and other container security activities since doing so allows teams to detect issues early in the build process. A number of free and open source libraries are still under development and will be released at continuous intervals to help businesses use them in real world scenarios. **END** 

 **By: Dr Gaurav Kumar**

The author is associated with various academic and research institutes for delivering expert lectures and conducting technical workshops on the latest technologies and tools.

Docker SBOM: Keeping a Check on Security Risks

The Log4j vulnerability and SolarWinds supply chain attack have made us realise that software supply chains are at great risk of being targeted by attackers. The Docker SBOM helps detect these risks.



Software supply chain security comprises everything required to build, deliver and run software anywhere with confidence. It ensures that not only software but the infrastructure, operating systems, source code repositories, registries, developers who wrote it, and the cloud services it uses don't pose any threats.

The Software Bill of Materials (SBOM) has emerged as the first step to ensure software security.

What is SBOM and how does it help businesses?

An SBOM is a list of software components that build the software product. It details the software package, content, licence, and certificate information, just like packaged food items contain ingredients and other information.

SBOMs help organisations know and keep track of all software components along with other details, such as:

- Determining all the runtime dependencies of the software
- Keeping track of critical updates and patches
- Reducing supply chain attacks and ensuring the organisation doesn't use vulnerable components
- Detecting licence and certificate-related issues

The National Telecommunications and Information Administration (NTIA) has listed the minimum requirements for the SBOM, which are:

- Software supplier and author of SBOM
- Information about software vendors
- Software component name and its version
- Unique identifier for each component

- Relationship of the component with other components and packages
- Time stamp of SBOM creation

SBOM standards

To make organisations adopt SBOM quickly, several standards have been defined for generating them. A defined schema provides a format for listing software components, and the most widely used are Software Package Data Exchange (SPDX) and CycloneDX.

SPDX: SPDX is an SBOM format developed by the Linux Foundation. It shows relationships between software components, licences, and copyrights associated with the software. It can be stored in different file formats, such as SPDX and JSON. It defines three elements: documents (metadata about the SBOM), packages (groups of elements), and files (single files).

CycloneDX: This lightweight SBOM standard is maintained by the OWASP community and built on top of SPDX. It gives security-related information such as supplier, manufacturer, target components, etc. It can be stored in different file formats, such as XML and JSON. It gets divided into components, services, and dependencies.

Docker, too, has recently introduced an SBOM format for container images. It has announced the *docker sbom* CLI command in Docker Desktop 4.7.0 as a CLI plugin, which lists all the components of the container image. It is an experimental functionality for now, which has been developed as an open source collaboration with Anchore through the Syft tool.

Syft is a tool for generating SBOMs from container images and file systems. It can convert SBOMs to different formats.

Installation of the Docker CLI plugin

- As a pre-requisite, Docker/Docker Desktop must be installed on your system. Upgrade the Docker desktop if its version is less than 4.7.0.
- Install the *docker sbom* plugin.

```
curl -sSfL https://raw.githubusercontent.com/docker/sbom-cli-plugin/main/install.sh | sh -s -
```

- Check its version and verify installation:

```
docker sbom version
```

Note: If you are using Docker as a CLI tool, then before installing *docker sbom* create a */root/.docker* directory. More information can be found at <https://github.com/docker/sbom-cli-plugin/issues/25>.

Different Docker SBOM commands

- To check all options related to *docker sbom*, type:

```
docker sbom -help
```

- To list the software components of an NGINX Docker image, type:

```
docker sbom nginx:latest
```

By default, the output will be in a table format. From the output given below, it can be inferred that the first image is pulled if not present locally, then it's loaded and processed further, and then the software elements get listed along with

```
root@master:~# docker sbom --help
Usage: docker sbom [OPTIONS] COMMAND

View the packaged-based Software Bill Of Materials (SBOM) for an image.

EXPERIMENTAL: The flags and outputs of this command may change. Leave feedback on https://github.com/docker/sbom-cli-plugin.

Examples:
  docker sbom alpine:latest          a summary of discovered packages
  docker sbom alpine:latest --format syft-json  show all possible cataloging details
  docker sbom alpine:latest --output sbom.txt  write report output to a file
  docker sbom alpine:latest --exclude /lib --exclude '**/*.db'  ignore one or more paths/globs in the image

Options:
  -D, --debug                show debug logging
  --exclude stringArray      exclude paths from being scanned using a glob expression
  --format string            report output format, options=[syft-json cyclonedx-xml cyclonedx-json github-json spdx-tag-value spdx-json table text] (default "table")
  --layers string            [experimental] selection of layers to catalog, options=[squashed all] (default "squashed")
  -o, --output string        file to write the default report output to (default is STDOUT)
  --platform string          an optional platform specifier for container image sources (e.g. 'linux/arm64', 'linux/arm64/v8', 'arm64', 'linux')
  -q, --quiet                suppress all non-report output
  -v, --version              version for sbom

Commands:
  version  Show Docker sbom version information

Run 'docker sbom COMMAND --help' for more information on a command.
```

Figure 1: Different sub-commands with *docker sbom*

its version and package type.

```
root@master:~# docker sbom nginx:latest
Syft v0.46.3
✓ Pulled image
✓ Loaded image
✓ Parsed image
✓ Cataloged packages
[143 packages]
NAME                VERSION                TYPE
adduser             3.118                 deb
apt                 2.2.4                 deb
base-files          11.1+deb11u5          deb
base-passwd         3.5.51                deb
bash                5.1-2+deb11u1         deb
bsdutils            1:2.36.1-8+deb11u1    deb
ca-certificates     20210119               deb
coreutils           8.32-4+b1             deb
curl                7.74.0-1.3+deb11u3    deb
dash                0.5.11+git20200708+dd9ef66-5 deb
debconf             1.5.77                 deb
debian-archive-keyring 2021.1.1               deb
debianutils         4.11.2                 deb
diffutils           1:3.7-5                deb
dpkg                1.20.12                deb
. . .
.
ncurses-bin         6.2+20201114-2        deb
nginx               1.23.2-1~bullseye     deb
nginx-module-geoip  1.23.2-1~bullseye     deb
nginx-module-image-filter 1.23.2-1~bullseye     deb
nginx-module-njs    1.23.2+0.7.7-1~bullseye deb
nginx-module-xslt   1.23.2-1~bullseye     deb
openssl             1.1.1n-0+deb11u3       deb
passwd              1:4.8.1-1              deb
perl-base          5.32.1-4+deb11u2       deb
readline-common     8.1-1                  deb
sed                 4.7-1                  deb
sensible-utils      0.0.14                 deb
sysvinit-utils      2.96-7+deb11u1         deb
tar                 1.34+dfsg-1            deb
tzdata              2021a-1+deb11u8        deb
ucf                 3.0043                 deb
util-linux          2.36.1-8+deb11u1       deb
zlib1g              1:1.2.11.dfsg-2+deb11u2 deb
```

- To list the software components of a Node.js application, type:

```
docker sbom oshi36/nodeapp_test:latest
```

```
root@master:~# docker sbom oshi36/nodeapp_test:latest
Syft v0.46.3
✓ Loaded image
✓ Parsed image
✓ Cataloged packages [569 packages]
NAME                VERSION                TYPE
JSONStream          1.3.5                 npm
abbrev              1.1.1                 npm
accepts             1.3.8                 npm
agent-base         4.2.1                 npm
agent-base         4.3.0                 npm
agentkeepalive      3.5.2                 npm
ajv                 5.5.2                 npm
. . .
. . .
wrappy              1.0.2                 npm
write-file-atomic   2.4.3                 npm
xdg-basedir         3.0.0                 npm
xtend               4.0.1                 npm
y18n                3.2.1                 npm
y18n                4.0.0                 npm
yallist             2.1.2                 npm
yallist             3.0.3                 npm
yargs               11.1.1                npm
yargs-parser        9.0.2                 npm
yarn                1.22.4                npm
zlib                1.2.11-r3             apk
```

- To write the SBOM report to a file, use the *output* flag:

```
docker sbom oshi36/nodeapp_test:latest --output nodeapp_report
```

Open the output file *nodeapp_report*:

```
cat nodeapp_report
```

- To skip paths from *docker sbom* to scan, use the *exclude* flag:

```
docker sbom oshi36/nodeapp_test:latest --exclude /usr/local
```

From the output given below, it can be seen that *docker sbom* leaves the elements to be scanned at the */usr/local* location.

```
docker sbom oshi36/nodeapp_test:latest --exclude /usr/local
Syft v0.46.3
✓ Loaded image
✓ Parsed image
✓ Cataloged packages [132 packages]
NAME                VERSION                TYPE
```



```

"tools": [
  {
    "vendor": "anchore",
    "name": "syft",
    "version": "v0.46.3"
  }
],
"component": {
  "bom-ref": "3c49c5c51b864d3e",
  "type": "container",
  "name": "oshi36/nodeapp_test:latest",
  "version": "sha256:d1e152be03b2b120ffd969eb276d59876ae6
e7e91510915d6080f0b5c36f1f02"
}
},
. . . . .
docker sbom oshi36/nodeapp_test:latest --format syft-json

```

It outputs the SBOM report in Syft's native JSON format:

```

{
  "artifacts": [
    {
      "id": "2a0a99c2a38a6407",
      "name": "JSONStream",
      "version": "1.3.5",
      "type": "npm",
      "foundBy": "javascript-package-cataloger",
      "locations": [
        {
          "path": "/usr/local/lib/node_modules/npm/node_modules/
JSONStream/package.json",
          "layerID": "sha256:8ca4f4055a7095ff1807ce521287ae7bdd91a
a711ae45d5b0f3d1c45ab0adefe"
        }
      ],
      "licenses": [
        "(MIT OR Apache-2.0)"
      ],
      "language": "javascript",
      "cpes": [
        "cpe:2.3:a:JSONStream:JSONStream:1.3.5:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:*:**",
        "cpe:2.3:a:*:JSONStream:1.3.5:*:*:*:*:*:*:*:*:**"
      ],
      "purl": "pkg:npm/JSONStream@1.3.5",
      "metadataType": "NpmPackageJsonMetadata",
      "metadata": {
        "name": "JSONStream",
        "version": "1.3.5",
        "author": "Dominic Tarr <dominic.tarr@gmail.com> (http://

```

```

bit.ly/dominictarr)",
  "licenses": [
    "(MIT OR Apache-2.0)"
  ],
  . . . . .

```

Log4j 2 vulnerability use case

Log4j 2 is a logging library by Apache and its vulnerability CVE-2021-44228 allows remote code execution from a context that is easily available to an attacker. This vulnerability has been found in Minecraft servers. It allows commands to be typed in chat logs and then sent to the logger.

Currently, CVE-2021-44228 impacts all versions of Log4j 2.x <= 2.14.1. If the container image has Log4j, then *docker sbom* can help in finding out its version and deciding whether it's vulnerable or not.

- To list the components of jboss/wildfly application server and find whether its Log4j version is vulnerable or not, type:

```
docker sbom jboss/wildfly | grep log4j
```

From the output given below it can be seen clearly that Log4j is vulnerable to CVE-2021-44228 due to its version:

```

docker sbom jboss/wildfly | grep log4j
Syft v0.46.3
✓ Pulled image
✓ Loaded image
✓ Parsed image
✓ Cataloged packages      [1151 packages]
log4j-api                  2.14.1                      java-archive
log4j-jboss-logmanager    1.2.2.Final                 java-archive
log4j2-jboss-logmanager   1.0.0.Final                 java-archive

```

To sum up, *docker sbom* helps in generating the SBOM for the container image, which helps identify and solve any dependency issues. **END** 🐧

By: Oshi Gupta
 Oshi Gupta works as a DevOps engineer and technical writer at CloudYuga Technologies. She loves writing blogs and is keen to learn about various cloud native technologies.

The Benefits of Containerization

Containerization offers organisations a range of benefits including scalability and security. There are a number of open source platforms that offer containerization solutions. Some of the best are listed in this article.

Containerization has significantly picked up pace in the last couple of years. The catchphrase of containerization, ‘build once and deploy anywhere’, is the core essence of modern applications. However, how and to which area of the technology landscape it’s applied has evolved all the way from the operating system to applications to services – a.k.a. microservices.

Containerization has become the *de facto* industry standard for its portability, scalability, faster deployment, high productivity, enhanced security, continuity, and easy management. Various containerization products are available in the open source and proprietary software market to best suit a given organisation. Identifying the right product and implementing it correctly helps

unleash the power of containers. The best way to go about containerization is to identify the core requirements of the organisation and check the solution for portability, scalability, faster deployment, high productivity, enhanced security, continuity and easy management.

Kubernetes has become the industry leader in containerization solutions. Also widely adopted are Dockers,



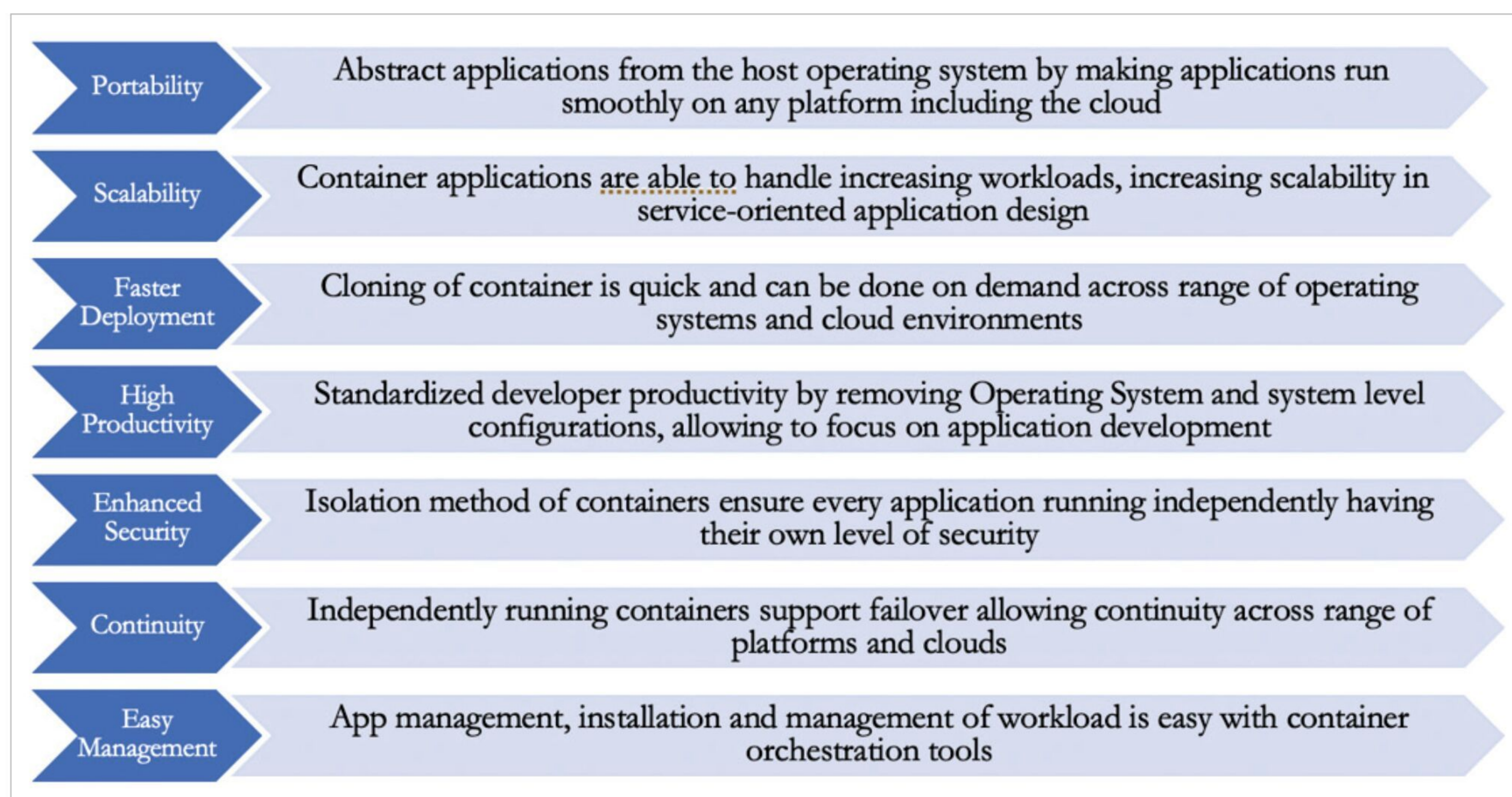


Figure 1: Benefits of containerization

Apache Mesos and OKD. Though there are many commercial container solutions, open source solutions are not only free but have rich features and a wide developer adoption base. Let's look at a few of them briefly, and the unique capabilities and functionality they offer.

Kubernetes is a Linux based open source container solution for cluster computing. The ease of managing organised applications is a key allure for developers and DevOps engineers. Creating an app, and specifying a pipeline and type of workload is what it takes for Kubernetes to automatically run in an environment. This feature helps Kubernetes score over traditional container-based applications with respect to agility and reliability. Notification is built in for monitoring application deployment and execution statistics, making it easy for developers to focus on scale and performance. Once scale and security are addressed to the needs of the organisation, Kubernetes runs on autopilot for container-based applications to grow into whatever environment organisations choose, including the cloud.

Docker is yet another popular and widely adopted containerization technology, which works well on the Kubernetes platform. The speed at which containers can be created is one of the key benefits of Docker.

Apache Mesos is an open source platform for automating deployment, scaling, and management of large clusters of container based applications. Ease of customisation for a wide range of applications is a significant differentiator of Mesos. The ability to run Dockers and other such programs across the distributed network is one of the reasons many organisations adopt Mesos.

OKD is an open source alternative to OpenShift with similar features and functionality. One of its top benefits is the ability to automatically distribute Kubernetes on all popular cloud platforms (AWS, Azure, GCP) and bare metal, OpenStack, Red Hat, and other virtualization providers. Quick application build, scaling, support for automatic high availability, load balancing, health check, and failover are the features that attract organisations to OKD.

In addition to these tools, there is a significant amount of innovation in containerization simplification in terms of maintenance and providing an additional layer of robustness. A good example is Toradex, which offers rich features when it comes to seamless integration with hardware, operating system, development tools, and remote management including updates and monitoring. The platform is highly reliable and secure. Toradex looks to increase efficiency across the board in the software development life cycle by providing easy-to-maintain products, and integration over-the-air including updates and device monitoring. This allows development teams to focus on what's important to the business. Its operating system Torizon takes advantage of the large open source ecosystem to integrate Docker Hub and thousands of ready-to-use containers. This open source platform is customisable, and can be deployed on different hardware. And it's built on open standards with open APIs to make integration with other open source and third party services

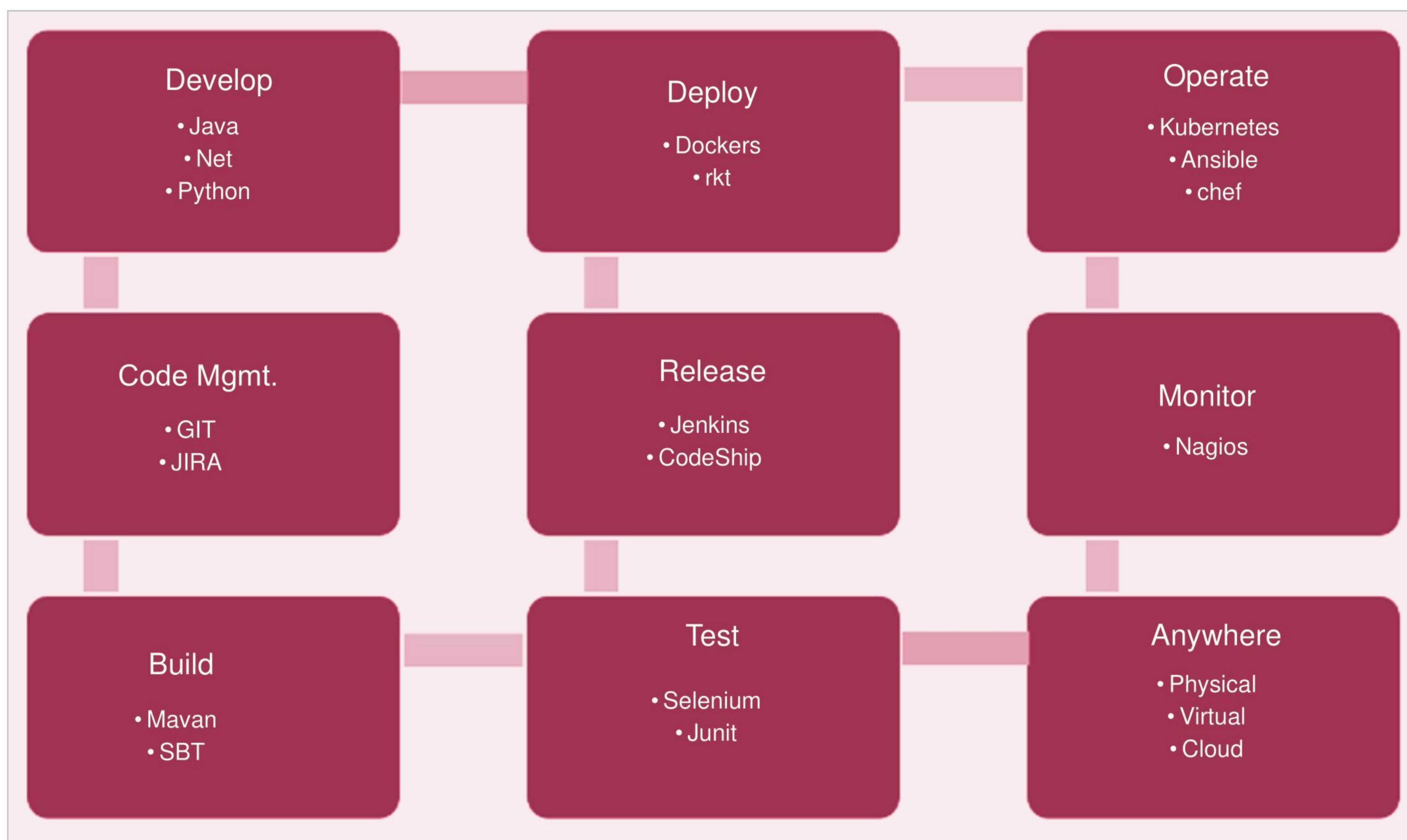


Figure 2: Application container DevOps reference architecture

easy. When an organisation is looking to develop IoT device based business solutions, using such tools helps to rapidly prototype and deploy them while the underlying platform takes care of the non-functional aspects of the application.

Containerization helps an organisation to identify the right tools and integrate them into an efficient process — from development and testing to deployment and operations. Figure 2 gives the reference architecture of container DevOps. As one can see, many of the tools are time-tested, popular, and widely adopted across the industry in development practices.

As the adoption of the cloud continues to evolve from cloud as a platform and cloud-native to cloud-agnostic and to a multi-cloud journey, the best technology strategy organisations can adopt is containerization. When containerization is established as a

platform with tools and technology that are a right fit for the organisation, it can be replicated across it for scalability, which is a major challenge enterprises face today.

Containerization, when adopted well in terms of application design, development and deployment, offers

simplified maintenance while allowing for an additional layer of robustness. There is a growing maturity in containerization tools, especially the open source ones. Organisations must identify the type of workload they are looking to containerize, and choose the right platform and tools. **END** 🐧

References

- What is containerization by RedHat, <https://www.redhat.com/en/topics/cloud-native-apps/what-is-containerization>
- Use containers to build, share and run your applications, <https://www.docker.com/resources/what-container/>
- Production-grade container orchestration, <https://kubernetes.io/>
- Nagios Monitoring, <https://www.nagios.com/>
- Toradex: Embedded computing made easy with containerization, <https://www.toradex.com/>
- Program against your datacenter like it's a single pool of resources, <https://mesos.apache.org/>

By: Bala Kalavala

The author is a seasoned emerging technical architect, evangelist, thought leader, and sought-after keynote speaker. He currently works as a distinguished member of the technical staff and head of the Enterprise Architecture practice as chief architect in a global technology consulting firm.

Podman:

The Pod Manager Tool



Podman (POD manager) is a daemon-less and open source Linux native tool by Red Hat for developing, managing, and running containers. It deploys applications using the Open Containers Initiative (OCI) container and container Images. Podman manages pods, containers, container images, etc, using the *libpod* library. Let's learn a few more things about it.

Podman uses the RESTful APIs to manage the containers. It has a remote Podman client, which is currently supported on Linux, Windows, and Mac whereas the RESTful service is only supported on Linux.

Installation of Podman

The instructions to install Podman on different operating systems are given in the table given on this page.

In this article we are going to install Podman on the Ubuntu operating system. The Podman package is available in the official repositories for Ubuntu 20.10 and above.

```
podman@osfy:~$ sudo apt-get -y update
podman@osfy:~$ sudo apt-get -y install podman
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

The following additional packages will be installed:

```
buildah catatonit common containernetworking-plugins crun
fuse-overlayfs golang-github-containernetworking-plugin-
dnsname golang-github-containers-common
golang-github-containers-image libostree-1-1 uidmap
.
.
<Out put snipped>
.
.
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
Processing triggers for man-db (2.10.2-1) ...
podman@osfy:~$
```

Let's verify the Podman installation using the *info* command:

```
podman@osfy:~$ podman info
```

The output of the command will provide you with all the

MacOS	Windows	Linux distributions
Podman machine is backed by a QEMU-based virtual machine.	Podman machine is backed by a virtualised Windows system for Linux (WSLv2) distribution.	Arch Linux and Manjaro Linux, Alpine Linux, CentOS, Debian, Fedora, Fedora-CoreOS, Fedora SilverBlue, Gentoo, OpenEmbedded, openSUSE, Raspberry Pi OS arm64 (beta), RHEL7, RHEL8, Ubuntu distributions are supported.
Use the terminal to execute Podman commands.	Use the Windows PowerShell (or CMD) prompt to execute Podman commands.	
Podman is provided through Homebrew.		
Give the brew install command to install Podman: <i>brew install podman</i>		
Create and start your first Podman machine: <i>podman machine init</i> <i>podman machine start</i>		
Verify the installation: <i>podman info</i>		

details about the host machine, hostname, kernel, memory, version and plugins.

You can use:

```
podman --help
```

...command to get help for different commands/parameters (Figure 1).

Once we have verified the details using *info* command, let's try and run the *nginx* container using Podman.

```
podman@osfy:~$ podman run -dt -p 8999:80/tcp docker.io/library/nginx
```

Trying to pull **docker.io/library/nginx:latest...**

```
Getting image source signatures
Copying blob a38226fb7aba done
Copying blob a603fa5e3b41 done
Copying blob c39e1cda007e done
Copying blob 9802a2cfd8d done
Copying blob 62583498bae6 done
Copying blob 90cfefba34d7 done
Copying config 88736fe827 done
Writing manifest to image destination
Storing signatures
```

```
685a357ad49fd2e7cd493abc6579aaa
9d2199f863b81ef7a389dfbad2d1b6219
```

Once the above command is successfully completed, we can access *localhost:8999* in the browser. It should display the NGINX home page as shown in Figure 2.

We can check the details of all existing containers by running the command *podman ps*. It should have the same container ID that was created earlier.

```
podman@osfy:~$ podman ps
CONTAINER ID IMAGE COMMAND
CREATED STATUS PORTS NAMES
685a357ad49f docker.io/library/nginx:latest nginx -g daemon
o... 6 minutes ago Up 6 minutes ago 0.0.0.0:8999->80/tcp
priceless_bose
```

You can also check the logs for NGINX containers if you need to debug anything. All you have to do is to run the command given below:

```
podman@osfy:~$ podman logs -l
```

Here, *-l* represents the latest container.

Now, let's stop the container using the *stop* command. The

```
Available Commands:
attach      Attach to a running container
auto-update Auto update containers according to their auto-update policy
build       Build an image using instructions from Containerfiles
commit      Create new image based on the changed container
container   Manage containers
cp          Copy files/folders between a container and the local filesystem
create      Create but do not start a container
diff        Display the changes to the object's file system
events      Show podman events
exec        Run a process in a running container
export      Export container's filesystem contents as a tar archive
generate    Generate structured data based on containers, pods or volumes
healthcheck Manage health checks on containers
help        Help about any command
history     Show history of a specified image
image       Manage images
images      List images in local storage
import      Import a tarball to create a filesystem image
info        Display podman system information
init        Initialize one or more containers
inspect     Display the configuration of object denoted by ID
kill        Kill one or more running containers with a specific signal
load        Load image(s) from a tar archive
login       Login to a container registry
logout      Logout of a container registry
logs        Fetch the logs of one or more containers
machine     Manage a virtual machine
manifest    Manipulate manifest lists and image indexes
mount       Mount a working container's root filesystem
network     Manage networks
pause       Pause all the processes in one or more containers
play        Play containers, pods or volumes from a structured file
pod         Manage pods
port        List port mappings or a specific mapping for the container
ps          List containers
pull        Pull an image from a registry
push        Push an image to a specified destination
rename      Rename an existing container
restart     Restart one or more containers
rm          Remove one or more containers
rmi         Removes one or more images from local storage
run         Run a command in a new container
save        Save image(s) to an archive
search      Search registry for image
```

Figure 1: Podman commands



Figure 2: *nginx* home page

output shows a running container, which is stopped using the *podman stop -l* command.

```
podman@osfy:~$ podman ps -a
CONTAINER ID IMAGE COMMAND
CREATED STATUS PORTS NAMES
99a2890c1327 docker.io/library/nginx:latest nginx -g daemon
o... 9 minutes ago Created 0.0.0.0:8080->80/tcp
upbeat_merkle
685a357ad49f docker.io/library/nginx:latest nginx -g daemon
o... 8 minutes ago Up 8 minutes ago 0.0.0.0:8999->80/tcp
priceless_bose
```

```
podman@osfy:~$ podman stop -l
685a357ad49fd2e7cd493abc6579aaa9d
```

```
2199f863b81ef7a389dfbad2d1b6219
```

```
podman@osfy:~$ podman ps -a
CONTAINER ID   IMAGE                                COMMAND
CREATED       STATUS      PORTS
NAMES
99a2890c1327  docker.io/library/nginx:latest      nginx -g daemon
o... 11 minutes ago Created          0.0.0.0:8080-
>80/tcp upbeat_merkle
685a357ad49f  docker.io/library/nginx:latest      nginx -g daemon
o... 10 minutes ago Exited (0) 3 seconds ago 0.0.0.0:8999-
>80/tcp priceless_bose
```

We can also remove the container using the `podman rm` command. Here, `-l` represents the latest container as

mentioned earlier. Once you have removed the container, the output of `podman ps -a` does not show any result.

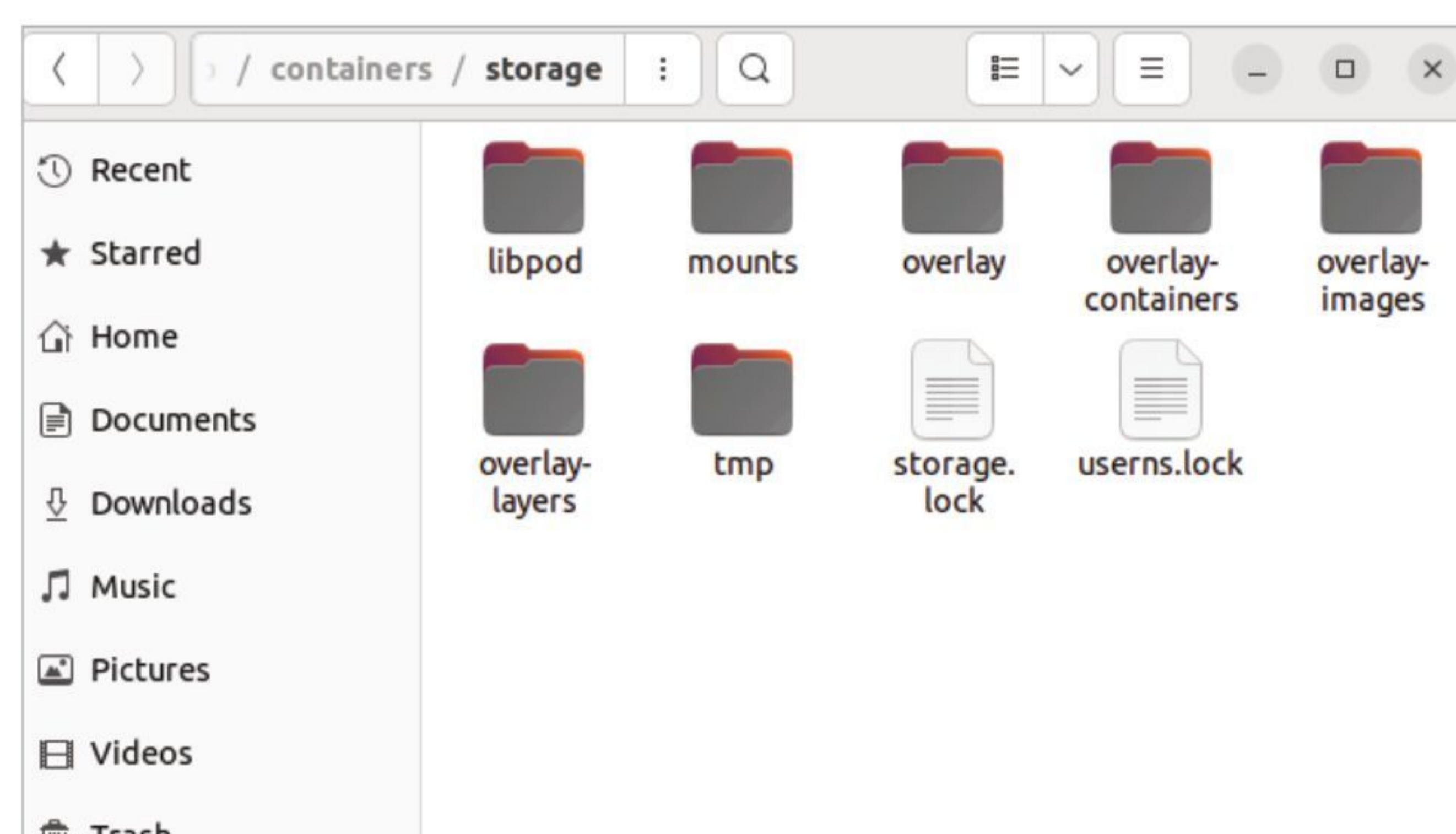


Figure 3: Container storage location

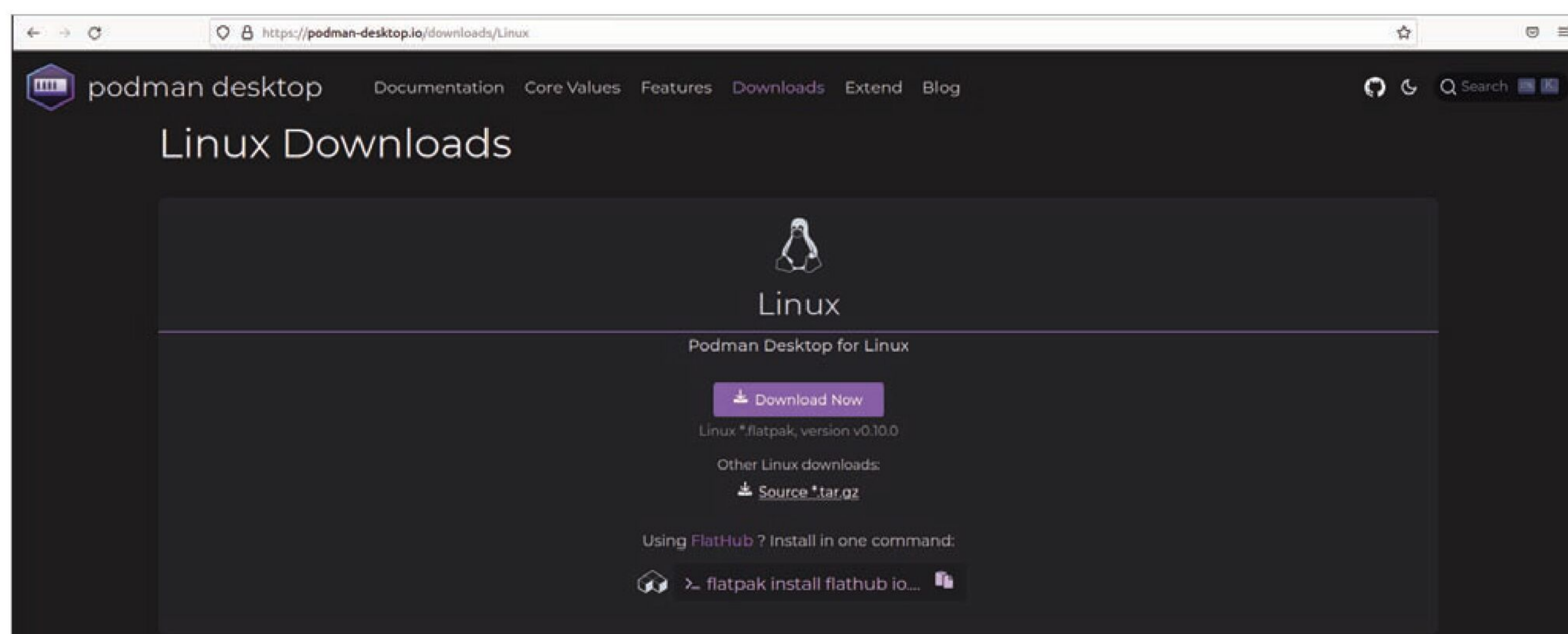


Figure 4: Podman Desktop home page

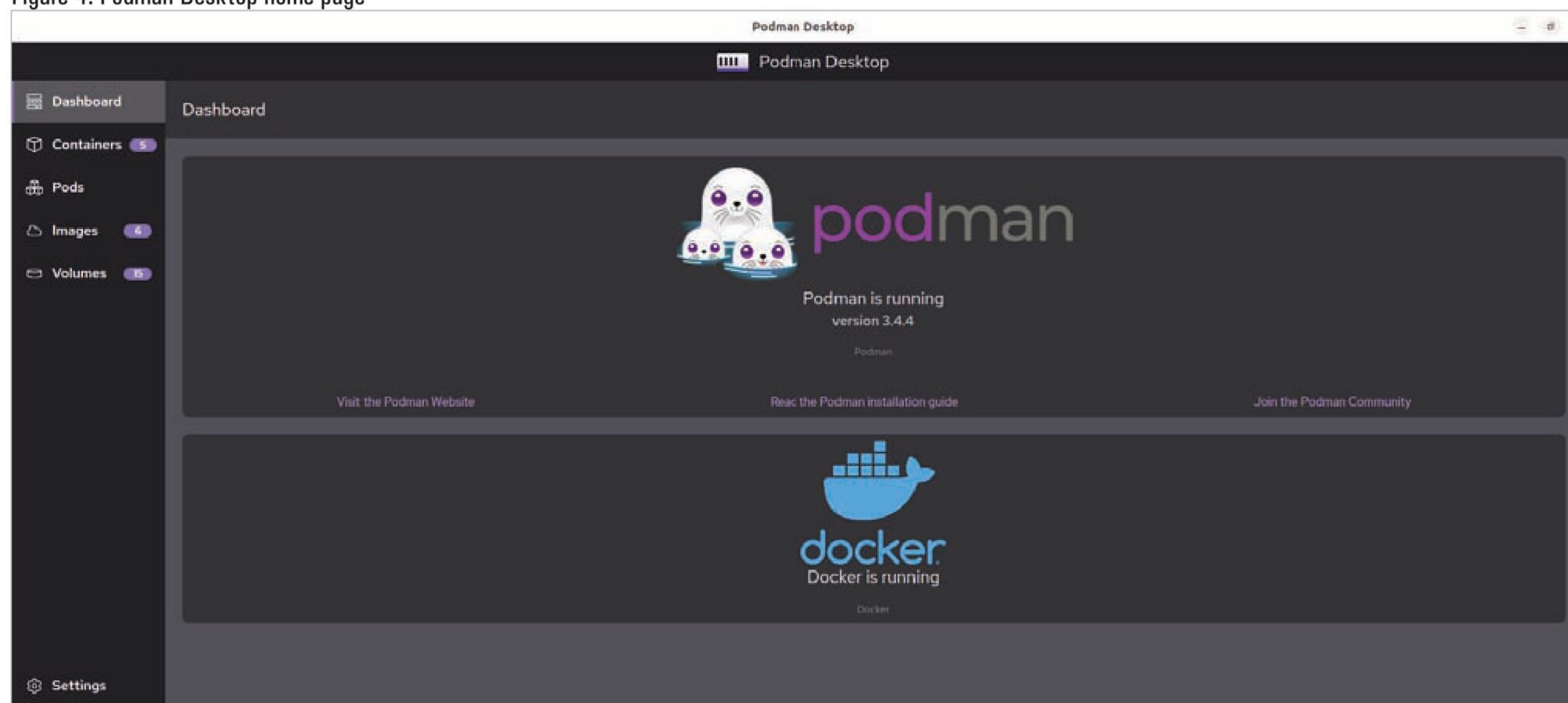


Figure 5: Podman and Docker status

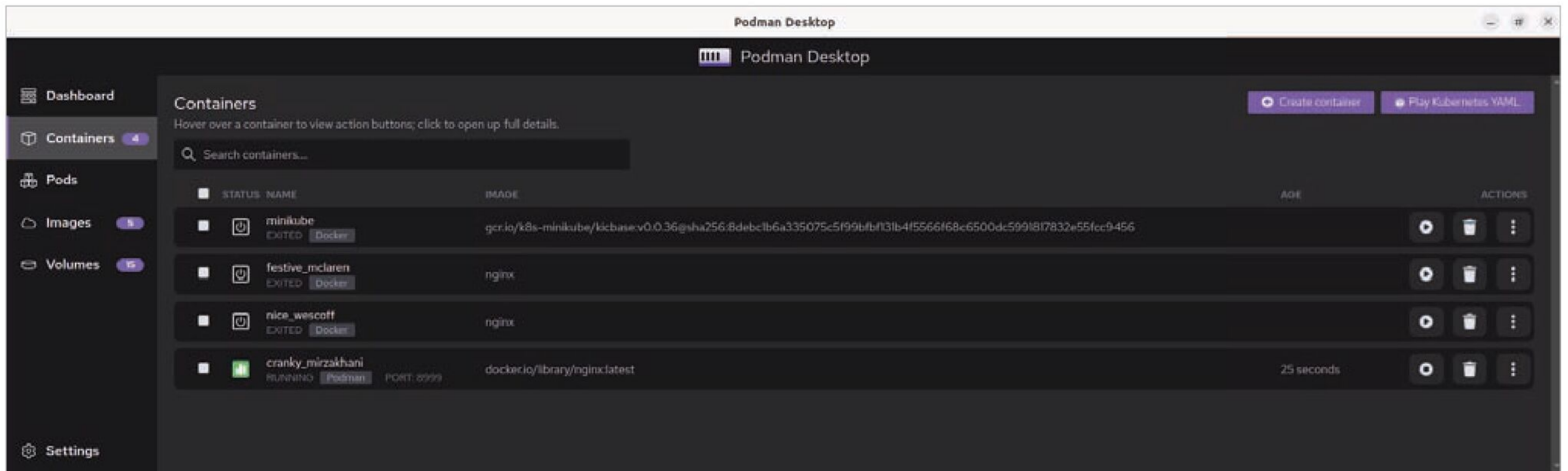


Figure 6: Containers available

```
podman@osfy:~$ podman rm -l
685a357ad49fd2e7cd493abc6579aaa9d2199f863b
81ef7a389dfbad2d1b6219
podman@osfy:~$ podman rm -l
99a2890c132723d18f34394e84a12bc9b3fc63e83bb
cbf7c0c185cf8b84f8ae0
podman@osfy:~$ podman ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES
```

You can find all container-related information at `/home/<user>/.local/share/containers/storage`, as shown in Figure 3.

Podman vs Docker

We have seen how Podman works and tried a few simple commands. The table above lists the differences between the two.

Podman Desktop

Podman Desktop is an open source GUI tool. It helps you to work with containers and Kubernetes. Visit <https://podman-desktop.io/> to download the Podman Desktop installation package.

Extract the installation package into your local system and click on the Podman desktop icon.

Once you do that, it will open the screen where you can see the status of Podman and Docker (Figure 5).

The container section will show all the available containers, as seen in Figure 6.

In this article, we have learned about Podman, its commands, Podman Desktop, and the difference between

	Podman	Docker
Architecture	Podman has a daemon-less architecture.	Docker uses the daemon.
Approach	Podman uses the rootless approach.	Docker has also added rootless mode recently to its daemon configuration.
Root privilege	The container does not have root privilege by default, which makes Podman more secure than Docker.	Since daemons have root privileges in Docker, it is vulnerable to attack.
Image creation	Podman uses the Buildah tool to create container images.	Docker is self-sufficient to create container images.
Tools	It requires associated tools like Buildah and Skopeo.	It is an all-in-one tool.
Pod support	Podman supports pods with one or more containers.	Docker does not have the concept of a pod.
Desktop application	Podman Desktop	Docker Desktop

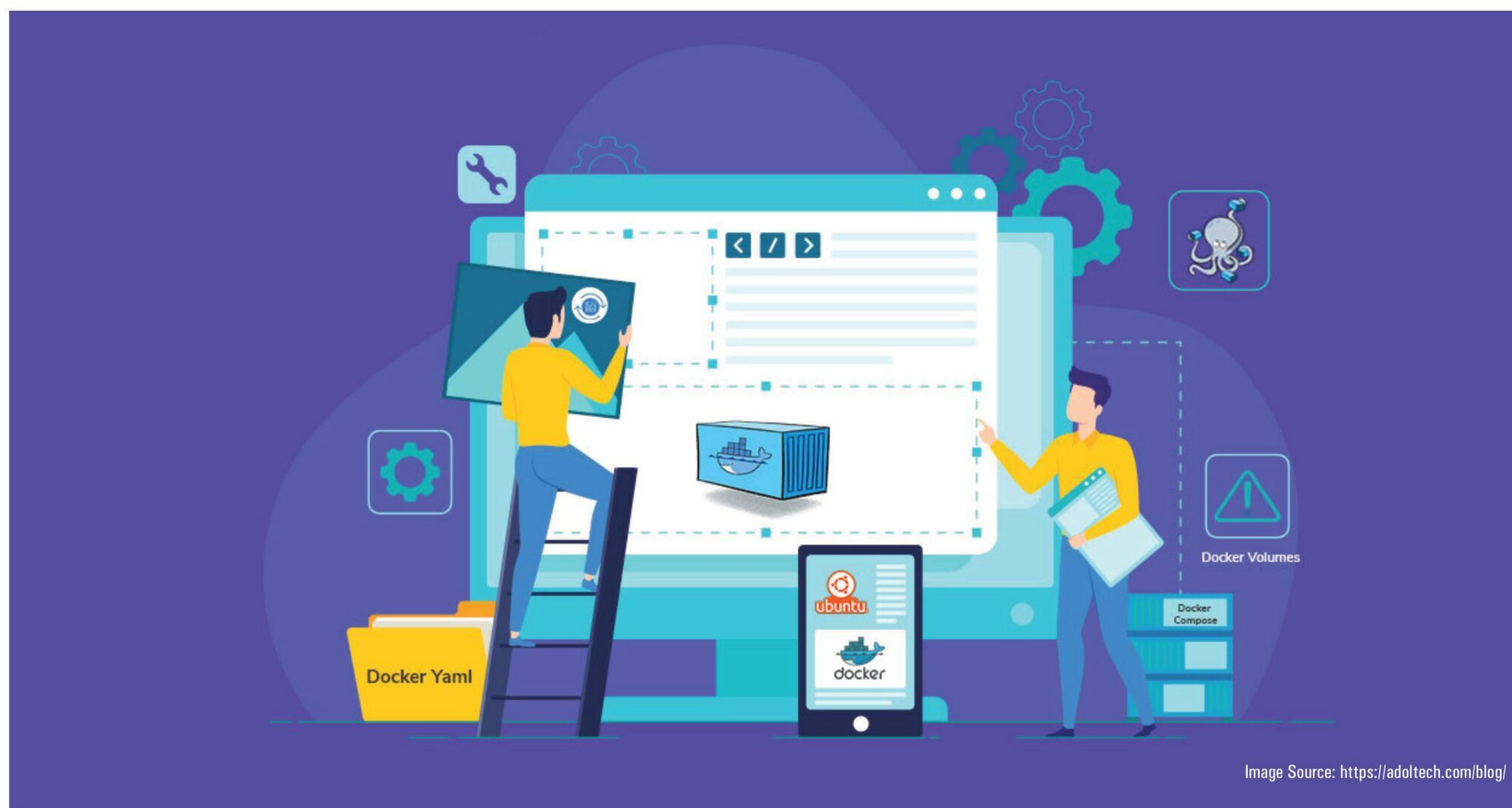
Podman and Docker. You should consider business needs and application goals as key factors while choosing the right tool. Happy learning!! END

- #### References
- What is Podman? <https://docs.podman.io/en/latest/>
 - Podman Overview, <https://www.redhat.com/en/topics/containers/what-is-podman#overview>
 - Podman Desktop, <https://podman-desktop.io/>

By: Ruby Verma

The author follows the motto 'Less talk more code', and likes to learn and explore new things. She has more than 12 years of experience in software development.

Setting Up a Docker Container



Docker provides a lightweight virtualization solution with zero overhead. This article demonstrates how to install Docker on Ubuntu. It also outlines the deployment of a Docker container application, NGINX, on Ubuntu.

Containerization is currently popular in the software development community as an alternative to virtualization. It enables us to encapsulate (containerize) software with all its dependencies and the runtime environment so that it can run uniformly across various infrastructures and platforms. In general, running software in containerized environments consumes lesser computing resources than running software in multiple virtual machines, because each virtual machine requires its own copy of the operating system. This article compares a virtual machine to a Docker container and demonstrates how to install and configure a Docker containerization environment. Finally, deployment of NGINX as a sample Docker container application is also demonstrated.

Virtual machine vs Docker container

Docker is a software development platform enabling virtualization on a single host with different operating systems. It enables rapid software delivery by separating infrastructure from applications. Docker virtualization is performed at the system level, using Docker containers,

unlike hypervisors, which produce VMs. As depicted in Figure 1, Docker containers operate the operating system of the host. This also improves an application's efficiency and security. Also, as containers consume fewer resources than virtual machines, we can deploy more containers on the same hardware.

A comparison of VM and Docker containers is given in Table 1.

Table 1: Comparison of virtual machine and Docker container

Virtual machine (VM)	Docker
Hardware-level isolation	OS-level isolation
Slow booting (minutes)	Fast booting (sec)
Heavyweight (Gigabyte)	Much smaller in size than a VM (KB/ MB)
VMs can simply migrate to new hosts	Containers are destroyed and rebuild
Huge computing resources required	Less computing resources required

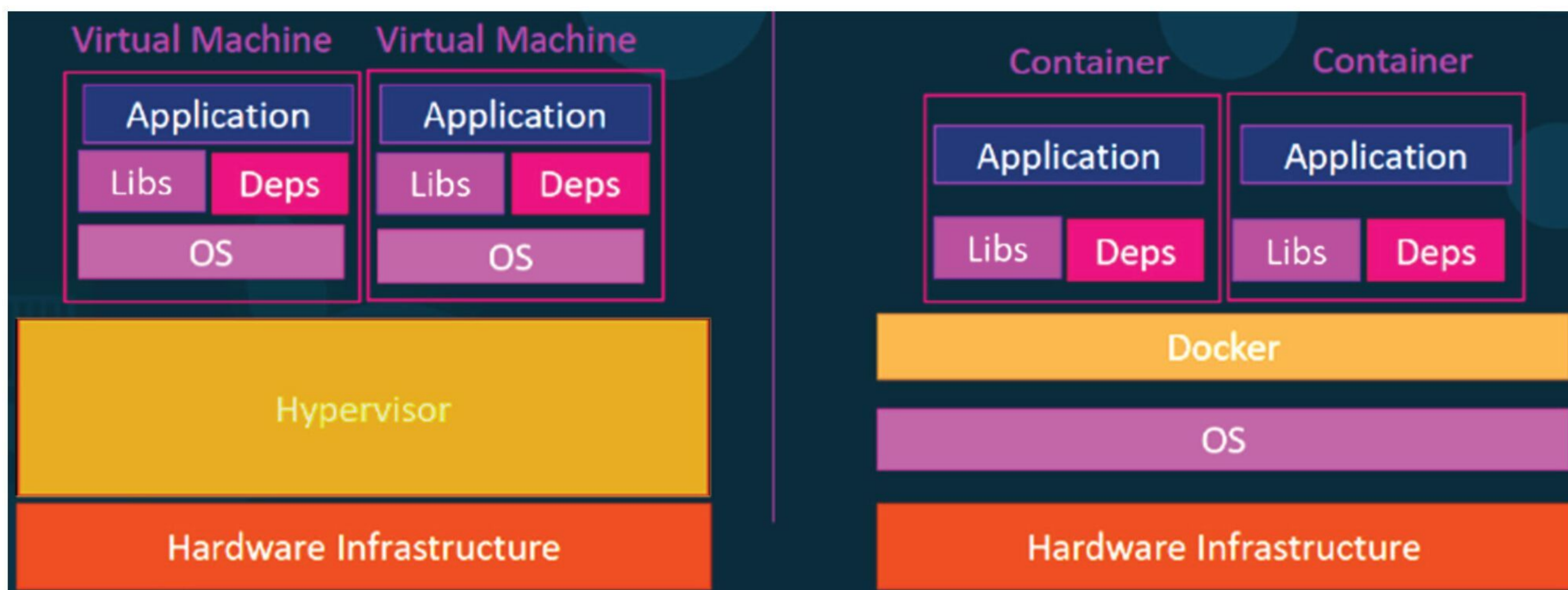


Figure 1: VM vs Docker container

Why Docker is popular

Docker provides a lightweight virtualization solution with zero overhead. It is currently widely used across a variety of sectors. Containers make it possible for developers to deploy a program along with all of its requirements and configurations, including libraries and other dependencies, as a single package, which can then be distributed to end users. Docker is gaining popularity in the world of technology for a variety of reasons, a few of which are mentioned below:

- Efficient resource usage
- Simple configuration
- Enhanced developer productivity
- Provides application isolation
- Fast application development

Installing Docker on Ubuntu/Linux

Before installing the Docker Engine on a new host system, we must configure the Docker repository. The user can then update and install Docker from the repository. The steps for installing Docker and running the example application are listed below.

Step 1: Perform an update on the package index for *apt*, and then install the necessary modules to enable *apt* to use HTTPS archives.

```
$ apt-get -y install apt-transport-https ca-certificates curl
```

Step 2: Download GPG key for Docker.

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Step 3: Configure the repository with the following command:

```
$ add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
```

Step 4: Once you have downloaded your *apt* package, install Docker Engine by executing the following commands:

```
$ apt-get update
$ apt-get -y install docker-ce
```

Step 5: Now verify the installation. Executing the Ubuntu image will allow you to confirm that the Docker Engine was successfully installed.

```
$ Docker pull ubuntu
```

Docker pulls a command to download the required image from the Docker hub, as shown in Figure 2.

Step 6: NGINX is free software used for many different purposes, including web serving, media streaming, reverse proxying, caching, and load balancing. Use the command given below to download the NGINX Docker container image:

```
$ Docker container run Nginx
```

```
root@k-master:/home/osboxes# docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
d51af753c3d3: Pull complete
fc878cd0a91c: Pull complete
6154df8ff988: Pull complete
fee5db0ff82f: Pull complete
Digest:
sha256:747d2dbbaaee995098c9792d99bd333c6783ce56150d1b11e333bbced5c54d7
Status: Downloaded newer image for ubuntu:latest
```

Figure 2: Pull an Ubuntu image from the Docker hub

```

root@docker-master:/home/osboxes# docker run nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
54fec2fa59d0: Pull complete
4ede6f09aeefe: Pull complete
f9dc69acb465: Pull complete
Digest: sha256:404ed8de56dd47adadadf9e2641b1ba6ad5c
Status: Downloaded newer image for nginx:latest

```

Figure 3: Creating NGINX as a sample Docker container application

When we run the above command, Docker looks for the image repository and downloads the latest version of NGINX. Thus, the NGINX container is created and gets a virtual IP inside the private bridge network depicted in Figure 3.

The following command will display a list of all currently active containers. Successful deployment of NGINX Docker containers can be accessed as shown in Figure 4.


```
$ docker ps
```

Table 2: Frequently used commands for Docker in a production environment

Command	Purpose
<code>\$ docker --version</code>	To find the version of Docker
<code>\$ "docker start "container_name"</code>	Start your container
<code>\$ "docker stop "container_name"</code>	Stop running container
<code>\$ docker kill "container_name"</code>	Kill a container
<code>\$ docker top "container_name"</code>	To list the running processes within a container
<code>\$ docker container stats "container_name"</code>	To get the CPU and memory usage by a container
<code>docker info</code>	Get information on the container



Figure 4: NGINX Docker container successfully installed

Docker can provide a containerization platform that bundles our application and all of its dependencies inside a Docker container to ensure that our application functions seamlessly in any environment. Containerization simply eliminates this problem by bundling (containerizing) the software code with the related configuration files, libraries, and other runtime dependencies. Thus, containers are much more lightweight than virtual machines, resulting in their widespread adoption across industries. **END** 

References

- <https://hub.docker.com/>
- <https://docs.docker.com/compose/compose-file/>

By: Dr Aditya Bhardwaj

The author is an M. Tech and PhD in CSE. He works as assistant professor in the School of Computer Science Engineering and Technology at Bennett University, Greater Noida. He has experience in cloud computing and open source technology.

OSFY Magazine Attractions During 2022-23

Month	Theme
March 2022	Security, Network Management and Monitoring
April 2022	Open Source Programming (Languages and tools)
May 2022	DevOps Special
June 2022	AI, Deep learning and Machine Learning
July 2022	Database management and Optimisation
August 2022	Cloud Special: Everything from management to implementation
September 2022	Mobile/Web App Development, Optimisation and Security
October 2022	Blockchain and Open Source
November 2022	Open Source and IoT and Edge
December 2022	Data science, Data security, Data Management, Storage and Backup
January 2023	Docker and containers
February 2023	Open Source on Windows and Best in the world of Open Source (Tools and Services)

Why Developers Opt for Containers

Open source containers are popular these days, and many tools are available for container-specific services. In this article, we will discuss containers, Kubernetes, and microservices from a developer's perspective.

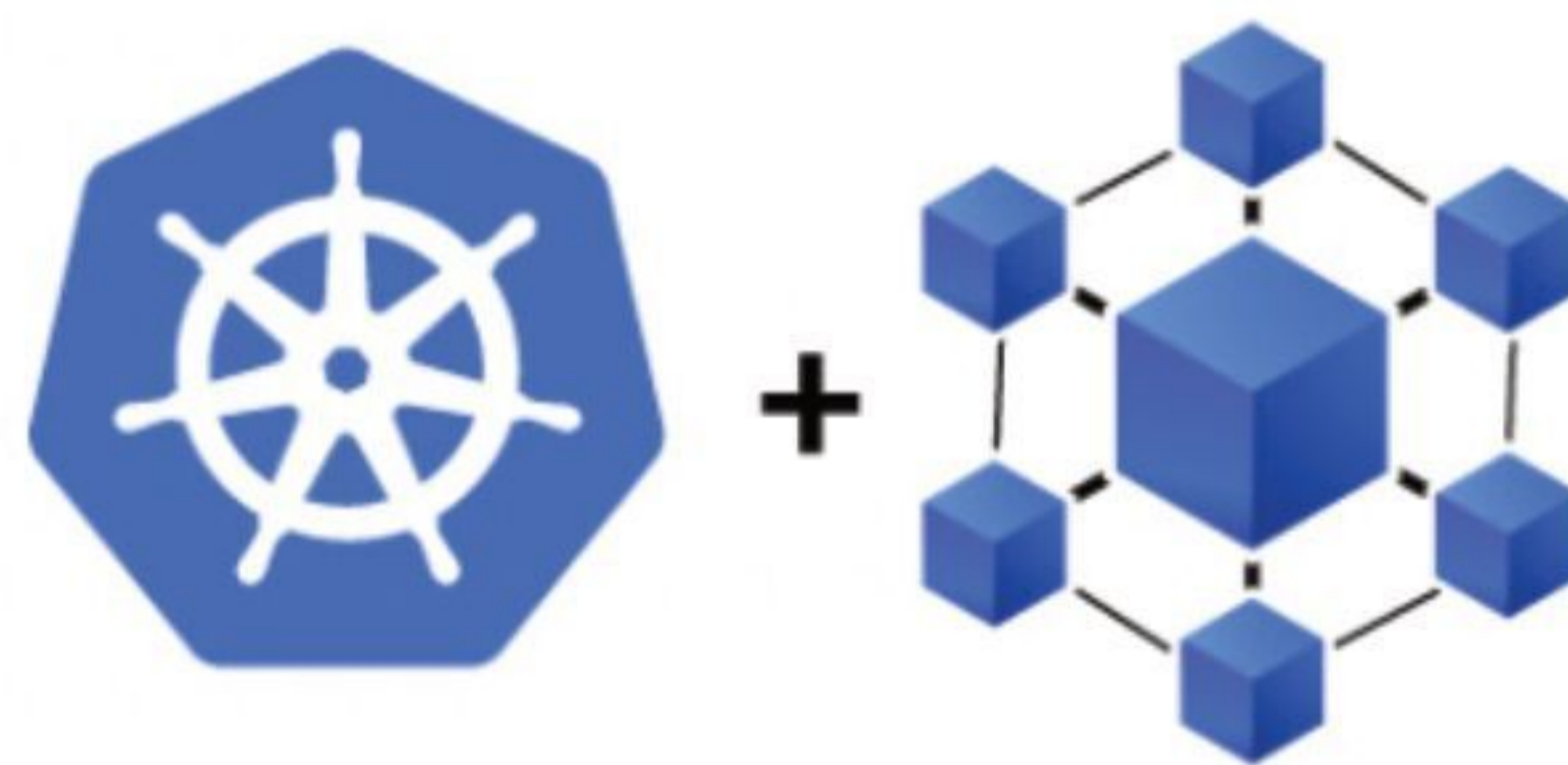
Containers are used by developers for deployment of packages in various operating systems. They are based on virtual machines, and have metadata and configuration for dependencies, packages, and properties files. Containers can execute packages on the operating system of the virtual machines. Development and deployment of applications become more efficient with containers and portable across different operating systems.

Containerization

Containerization is the process of executing code packages on the operating system. This happens without running the full virtual machine. The software packages and apps are deployed on containers. Containers can be run on the server or on the cloud-based compute node, and can talk to each other using networking protocols. The code package on containers has runtime configuration, code, tools, and dependencies. Each container has a runtime specific to the operating system. The code deployed is packaged to run the runtime of the vendor-specific container.

Popular containers

There are many popular containers like Docker, OpenShift, AWS Fargate, Google Kubernetes Engine, LXC, Portainer, Container Linux, and Apache Mesos. Many of these are open source and others are commercial



like Google, AWS, and Microsoft. These containers are helpful in DevOps and make posting updates and bug fixes easy. Applications can be scaled vertically and horizontally on the containers. They offer quick deployment, faster execution of different software apps, and scaling of the app without downtime.

“Containers are changing the data center the same way containers changed global trade.”

—*Jim Zemlin, executive director, Linux Foundation*

Kubernetes

Kubernetes is a popular open source container. The word is based on the Greek pilot or helmsman. Google Kubernetes Engine helps in operating different application containers on computing node clusters. Kubernetes is popular and distributed as GKE and EKS. EKS is the Amazon Elastic Container Service for Kubernetes. It helps in orchestrating different services and components across the computing nodes. Kubernetes provides a scalable infrastructure.

Use cases of Kubernetes

Google Cloud, Amazon Cloud, and Azure Cloud provide Kubernetes containers to deploy software packages and apps. The deployment and scaling can be automated. Apps can be ported across different operating systems, thus improving their efficiency and reliability. They can be deployed quickly and their upgradation is easier. Production-ready applications can be developed and tested in a short time on the cloud using Kubernetes.

The challenges

Developers do face a few issues while deploying software packages on Kubernetes. Applications require planning to be packaged on Kubernetes for deployment and for running them. Kubernetes is very popular as an orchestration tool for managing services, but developers face problems with its multiple distributions. To make the apps portable across distributions, developers need to go through a huge learning curve. Integration between different containers and Kubernetes is missing, which is a big issue.

Microservices with Kubernetes

Microservices architecture helps to design the app into small and independent services. Many apps are built using monolithic architecture, and companies are unable to manage and maintain them. Microservices architectural pattern helps to break down monolithic applications into

small services. Every microservice has the responsibility of providing the functionality. Developers can work on different microservices and test them independently before the QA engineers start doing the integration testing. In polyglot microservices architecture, microservices can be of different languages and stacks.

Microservices can be managed by a centralised service. Each microservice can be loosely coupled, and apps can be built and packaged to be deployed on the Kubernetes container. Kubernetes helps in deploying the distributed microservices based app and to post bug fixes and patches.

Containers have changed the way developers and test engineers deploy applications. Kubernetes has helped enterprises deploy apps on

“You need a highly automated deployment infrastructure. Ideally, you should use a deployment platform that provides the developer with a simple UI (command-line or GUI) for deploying and managing their services. The deployment platform will typically be based on virtual machines (VMs), containers, or serverless technology”

—*Chris Richardson in his book titled 'Microservices Patterns: With Examples In Java'*

different clouds and operating systems. It has simplified the development and management of cluster nodes. Automation of deployment with Kubernetes has accelerated the time

to market of applications. Next-gen Kubernetes will have better UX and reliable upgrading of the application. It can be designed to handle stateful applications. **END** 🐧

📧 **By: Bhagvan Kommadi**

The author is the founder of 'Quantica Computacao', India's first quantum computing startup. He has 20 years of experience in product development and IT services.

support@efy.in

Do you have a query, suggestion or a complaint?

You can e-mail it to support@efy.in and we will take care of the rest.

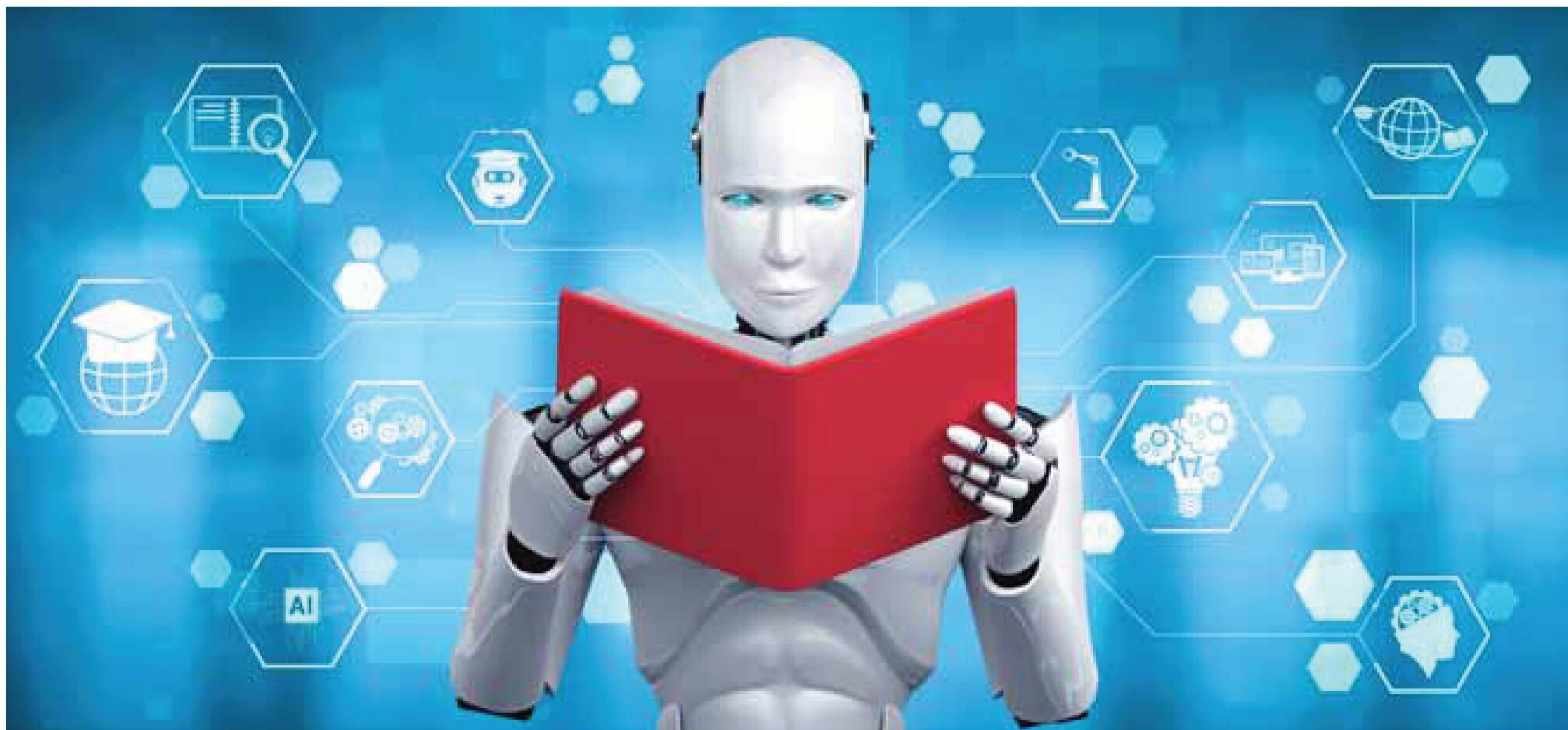


electronics
FOR YOU
www.electronicshobby.com


OpenSource
THE COMPLETE MAGAZINE ON OPEN SOURCE FOR YOU
www.opensourceforu.com

EFYGROUP
YOURS SINCE 1969
www.efy.in

AI: An Introduction to Scikit-learn and Our First Trained Model



In the last article in this series on AI and machine learning, we started our discussion of neural networks by using TensorFlow. We also got familiar with our first data set by using Keras. In this seventh article in the AI series, we will continue exploring neural networks and the use of data sets for training models. We will also introduce yet another powerful Python library for machine learning called scikit-learn. But we will begin by discussing two sensational applications that will show us the power of AI and machine learning.

penAI is an artificial intelligence (AI) research laboratory which does a lot of research in the fields of AI and machine learning. Elon Musk is one of the founding members of this organisation. In November 2022, the lab launched a new online tool called ChatGPT. It is an AI chatbot that can chat like a human being. This chatbot is a large language model trained with both supervised and reinforcement learning paradigms of machine learning. ChatGPT uses OpenAI's GPT-3.5 language model, an improved version of GPT-3 (Generative Pre-trained Transformer 3), a language model that uses deep learning to produce human-like text. I still feel the excitement I felt while using ChatGPT for the first time. The application clearly demonstrates the power of AI. The answers given by the chatbot are of high

quality and often indistinguishable from a reply given by a human being. You can use ChatGPT to correct grammatical errors, rephrase sentences, summarise passages, write programs, etc. Indeed, I have rephrased many of the sentences in this article using ChatGPT. Further, I have tested ChatGPT using text with intentional grammar mistakes, not the usual ones I commit, and the corrected sentences were highly accurate. The ability to re-phrase and summarise passages was also phenomenal.

It might be even possible that programmers could solve programming puzzles within a very short time using ChatGPT. One such claim was made in a recent programming contest called 'Advent of Code 2022'. In fact, in December 2022, just after one month of ChatGPT's existence, Stack Overflow banned its use for generating answers.

Figure 1 shows the Python program to add two matrices, written by ChatGPT. I asked for different programs in languages like BASIC, FORTRAN, Pascal, Haskell, Lua, Pawn, C, C++, Java, and even in some esoteric programming languages like Brainfuck, Ook!, etc, but ChatGPT always came up with an answer. I am pretty sure that ChatGPT is not copying the programs from some source on the Internet. Rather, I think ChatGPT is generating these answers by having a syntactic knowledge of the above-mentioned programming languages, obtained from the vast data with which it was trained. Many experts and observers are of the opinion that with the development of ChatGPT, AI has gone mainstream. Yet, the true power of ChatGPT is to be seen and analysed in the coming months and years.

Another wonderful AI based online tool provided by OpenAI is called DALL-E 2, named after WALL-E (a cartoon robot) and Salvador Dalí (a famous painter and artist). DALL-E 2 is an AI system that can create images and art from a description given in English. The tool will generate images in different styles like oil paintings, cartoons, caricatures, realistic paintings, surrealist paintings, mural paintings, etc. The tool can also imitate the styles of famous painters like Salvador Dalí, Pablo Picasso, Vincent van Gogh, etc. The images created by DALL-E 2 are of exceptionally high quality. I have tested the tool with the following description: 'Cubist painting of a happy man watching sunrise near a beach'. Figure 2 shows one of the images generated by DALL-E 2 from my description. Cubism is a painting style popularised by Pablo Picasso. Ask any painter friend of yours and he or she will confirm that this indeed is a painting drawn in the Cubist style. It was quite surprising to see that software, however complicated it may be, could imitate master craftsmen like

```

D Write a python program to add two matrices

Here is an example of a simple Python program that adds two matrices:

# Define the two matrices
A = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
B = [[10, 11, 12], [13, 14, 15], [16, 17, 18]]

# Initialize the result matrix
C = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]

# Add the matrices
for i in range(len(A)):
    for j in range(len(A[0])):
        C[i][j] = A[i][j] + B[i][j]

# Print the result matrix
print("Result matrix:")
for row in C:
    print(row)

This program uses a nested loop to iterate over the elements of the two matrices and calculate the sum. It then prints the result matrix to the screen.

```

Figure 1: A program written by ChatGPT



Figure 2: A Cubist painting generated by DALL-E 2

Picasso, van Gogh, Dalí, etc. I urge you to use this online tool. The experience will be quite entertaining as well as illuminating about the powers of AI. But remember that tools like ChatGPT and DALL-E 2 also bring up a lot of issues like copyright violations, copying of homework by students, etc.

Introduction to scikit-learn

Now, it is time for us to start using scikit-learn, a very powerful Python library for machine learning. This free and open-source software is licensed under the new BSD licence. Scikit-learn offers regression, classification, clustering and dimensionality reduction algorithms like support vector machines, random forests, k-means clustering, etc.

In this introductory discussion on scikit-learn, we will discuss code to implement support vector machines (SVM), a key idea in machine learning. An SVM is a supervised learning model. SVM is used for classification and regression analysis. The person behind the development of SVMs is a computer scientist called Vladimir Vapnik. He is a key figure in machine learning and,

together with Alexey Chervonenkis, he proposed the Vapnik–Chervonenkis (VC) dimension, a theoretical framework to measure the capacity of a classification model.

Now, consider the program shown in Figure 3. First, let us try to understand the code. The program uses SVM to classify data. Line 1 imports the `svm` module from `scikit-learn`. Notice that `scikit-learn` can be very easily installed by using `Anaconda Navigator`, as shown in the previous articles in this series. Line 2 defines a list called `X` which contains the training data. All the elements in `X` are lists of size 3. Line 3 defines a list `y`, which contains the class labels associated with the data in the list `X`. In this example, the data is classified into just two classes labelled 0 and 1. However, it is possible to classify data into multiple classes using the same technique. Line 4 uses the `SVC()` method provided by the module `svm`. `SVC()` method generates a support vector classifier. Line 5 uses the `fit()` method provided by the module `svm` to fit the SVM classifier model according to the given training data, in this case the arrays `X` and `y`. Finally,

```

1 from sklearn import svm
2 X = [[1,1,1], [2,2,2], [3,33,333], [4,44,444]]
3 y = [0,0,1,1]
4 clf = svm.SVC()
5 clf.fit(X, y)
6 print(clf.predict([[5., 5., 5.]])
7 print(clf.predict([[5., 55., 555.]])

[0]
[1]

```

Figure 3: Classification with SVM

```

1 from sklearn import svm
2 X = [[1,1], [2,2], [3,3], [4,4], [5,5]]
3 y = [100.0, 200.0, 300.0, 400.0, 500.0]
4 regr = svm.SVR()
5 regr.fit(X, y)
6 print(regr.predict([[1, 4]]))

[299.72775944]

```

Figure 4: Regression with SVM

lines 6 and 7 make predictions based on this classifier. The results of the predictions are also shown in Figure 3. Notice that the classifier is able to distinguish between the two sets of data we have provided for testing.

Now, let us see an example where SVM is used for regression. Consider the code shown in Figure 4. First, let us try to understand the code. Here again, Line 1 imports the `svm` module from `scikit-learn`. Line 2 defines a list called

`X`, which contains the training data. Notice that all the elements in `X` are lists of size 2. Line 3 defines a list `y` which contains the values associated with the data in the list `X`. Line 4 uses the `SVR()` method provided by the module `svm`. This method generates a support vector regression model. Line 5 uses the `fit()` method provided by the module `svm` to fit the SVM regression model according to

the given training data, in this case the arrays `X` and `y`. Finally, Line 6 makes a prediction based on this SVM regression model. The result of this prediction is also shown in Figure 4. There are two other methods in addition to `SVR()`, `LinearSVR()` and `NuSVR()`, to implement a support vector regression model. Replace Line 4 with the lines of code `'regr = svm.LinearSVR()'` and `'regr = svm.NuSVR()'`, and execute the code to

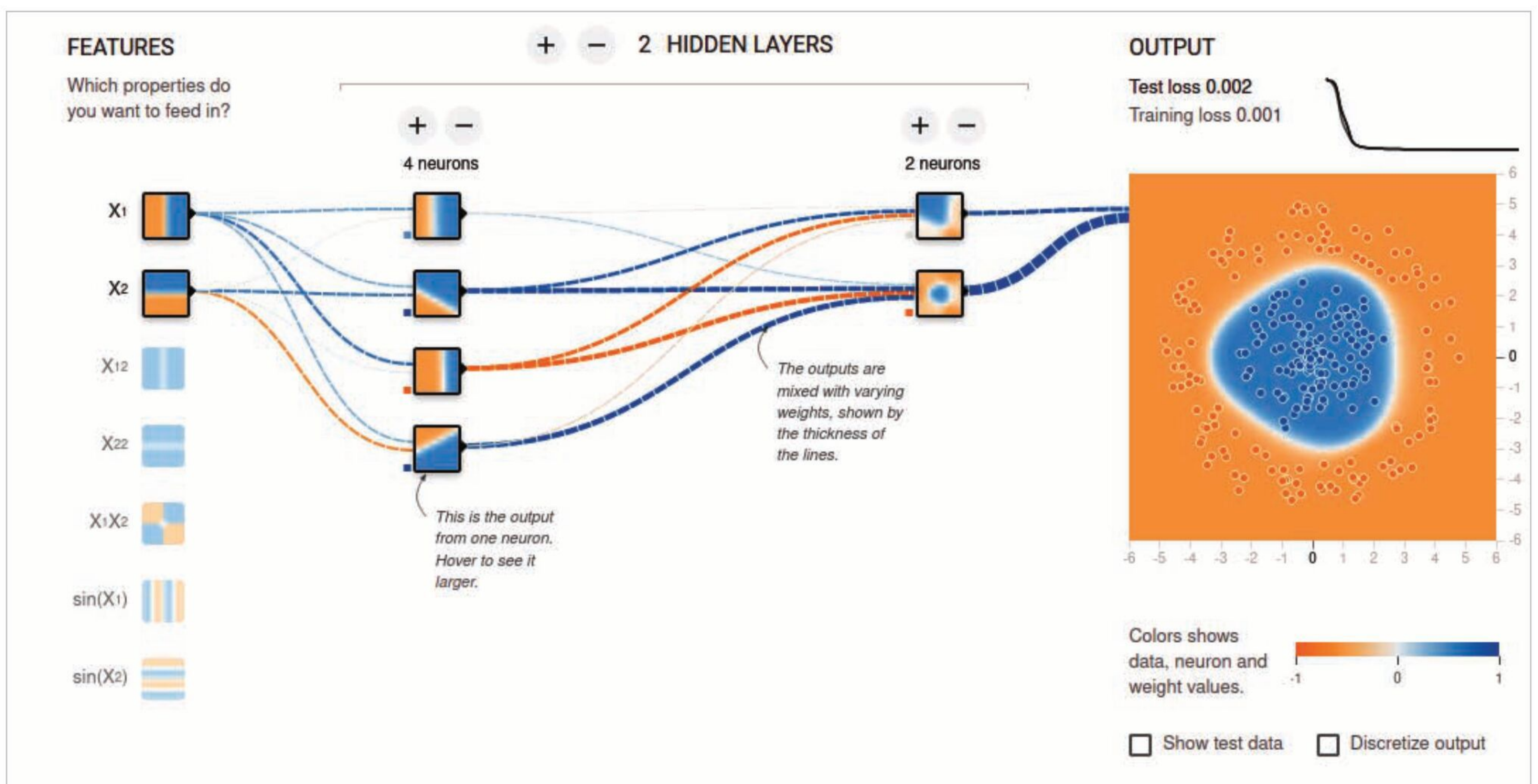


Figure 5: The neural network playground in action

see the effect of these support vector regression models also.

Let us now shift our focus to neural networks and TensorFlow. But we will definitely revisit other methods provided by scikit-learn in the next article in this series when we discuss unsupervised learning and clustering.

Neural networks and TensorFlow

The deployment of neural networks is the main strength of machine learning based applications. Let us go through a few more examples from the *nn* (primitive Neural Net Operations) module of TensorFlow to get better familiarity with the neural network jargon. We have already seen two activation functions called ReLU (rectified linear unit) and Leaky ReLU offered by the module *nn* of TensorFlow. There are other activation functions also offered by the module *nn*. The method *tf.nn.crelu()* computes the concatenated ReLU. The method *tf.nn.elu()* computes the exponential linear function. Soon, we will use one of these activation functions while training our first model using TensorFlow and Keras.

But before we move on to that section, let me share with you the details of an excellent visual tool that will help you understand the working of a neural network. This resource called 'Neural Network Playground' is provided by TensorFlow itself. You can visually add new neurons and new hidden layers to a neural network and then train that model. You can choose between activation functions like Tanh, Sigmoid, Linear and ReLU. Both classification and regression models can be analysed by using this tool. The effect of the training is shown as an animation. Figure 5 shows a sample neural network and the output provided by it. The tool is available online at the following URL, <https://playground.tensorflow.org>. This is a great resource for learning

about neural networks and this might be a good time in our journey through AI and machine learning to learn more about neural networks.

Our first trained model

Now, let us continue our discussion on the MNIST database of handwritten digits, a topic which we started to learn in the last article. In this article, we will train our model completely and test it with an image of a handwritten digit. The complete program, named *digit.py*, is relatively large. Hence, a part by part explanation is given here for better understanding. Line numbers are added to the program for ease of explanation.

```
1. import numpy as np
2. from tensorflow import keras, expand_
   dims
3. from tensorflow.keras import layers
4. num_classes = 10
5. input_shape = (28, 28, 1)
6. (x_train, y_train), (x_test, y_test)
   = keras.datasets.mnist.load_data( )
```

Lines 1 to 3 load the necessary packages and modules. Line 4 defines the number of classes as 10 because we are trying to classify the 10 digits (0 to 9). Line 5 defines the input shape as (28, 28, 1) because the data set we are going to use is a collection of 28 X 28 pixel grayscale images of handwritten digits. Line 6 loads this data set and divides it into training data and test data. More details about this data set were provided in the previous article in this series on AI and machine learning.

```
7.     x_train = x_train.
   astype("float32") / 255
8.     x_test = x_test.astype("float32")
   / 255
9.     x_train = np.expand_dims(x_
   train, 3)
10.    x_test = np.expand_dims(x_test,
   3)
11.    y_train = keras.utils.to_
   categorical(y_train, num_classes)
```

```
12.    y_test = keras.utils.to_
   categorical(y_test, num_classes)
```

Lines 7 and 8 change the pixel values of the image, which is currently in the range [0, 255], to the new range [0, 1]. The method *astype()* is used for typecasting integer values to floating-point values. Lines 9 and 10 expand the shapes of arrays *x_test* and *x_train* (changing the shape from (60000, 28, 28) to (60000, 28, 28, 1)). The lists *y_train* and *y_test* contain an array of digit labels denoting the classes. In this case, it is 10 denoting the 10 digits from 0 to 9. Lines 11 and 12 convert the lists *y_train* and *y_test* to binary class matrices.

```
13.    try:
14.        model = keras.models.load_
   model("existing_model")
15.    except IOError:
16.        model = keras.Sequential(
   [
       keras.Input(shape=input_shape),
       layers.Conv2D(32, kernel_
   size=(3, 3), activation="relu"),
       layers.MaxPooling2D(pool_
   size=(2, 2)),
       layers.Conv2D(64, kernel_
   size=(3, 3), activation="relu"),
       layers.MaxPooling2D(pool_
   size=(2, 2)),
       layers.Flatten( ),
       layers.Dropout(0.5),
       layers.Dense(num_classes,
   activation="softmax"),
   ]
   )
17.        batch_size = 64
18.        epochs = 25
19.        model.compile(loss="categorical_
   crossentropy", optimizer="adam",
   metrics=["accuracy"])
20.        model.fit(x_train, y_train,
   batch_size=batch_size, epochs=epochs,
   validation_split=0.1)
21.        model.save("existing_model")
```

Remember that model training is a processor intensive and memory consuming operation. Thus, with lines

13 and 14, we try to load a model from the directory *existing_model*, if such a model is already trained and saved there. The first time you execute this code, no model is existing and hence an exception will be raised. Line 15 handles this exception by executing lines 16 to 21 which define, train and save our model. The 16th line of code (which spans over multiple lines) defines our model. The parameters given in this line decide the behaviour of our model. We are using a sequential model having a stack of layers such that each layer has exactly one input tensor and one output tensor. Our discussion of the parameters used to define the model is deferred till the next article in this series. Since neural networks are so important in machine learning, our discussion then will cover a lot of details. Until then, consider the neural network used here as a black box.

Line 17 defines the batch size as 64. This line of code decides the number of samples per batch of computation. Line 18 defines the number of epochs as 25. This value decides the number of times the entire data set will be processed by the learning algorithm. Line 19 configures the model for training. Line 20 trains the model for the given number of epochs, which in this case is 25. The detailed explanation of these two lines of code is also deferred till the next article. Finally, Line 21 saves our trained model to the directory named *existing_model*. You will find the model saved in this directory as a number of *.pb* (TensorFlow) files. Notice that lines 16 to 21 are inside the *except* block.

```
22. print(model.summary( ))
23. score = model.evaluate(x_test, y_test, verbose=0)
24. print("Test loss:", score[0])
25. print("Test accuracy:", score[1])
```

Line 22 prints a summary about the model trained by us. Figure 6 shows this summary of the model trained.

Model: "sequential"		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 64)	0
flatten (Flatten)	(None, 1600)	0
dropout (Dropout)	(None, 1600)	0
dense (Dense)	(None, 10)	16010
=====		
Total params: 34,826		
Trainable params: 34,826		
Non-trainable params: 0		

Figure 6: Details of the model trained

Recall that the data set loaded in the beginning was divided into training data and test data. Line 23 uses the test data to test the accuracy of the model we have trained. Lines 24 and 25 print the details of this test.

```
26. img = keras.utils.load_img("sample1.png").resize((28, 28)).
    convert('L')
27. img = keras.utils.img_to_array(img)
28. img = img.reshape((1, 28, 28, 1))
29. img = img.astype('float32')/255
30. score = model.predict(img)
31. print(score)
32. print("Number is",
    np.argmax(score))
33. print("Accuracy", np.max(score) *
    100.0)
```

Now, it is time for us to deploy our model. I have written a few digits on paper and scanned them so that our model can use these images for classification. Figure 7 shows one of the images I have used to test our model. Line 26 loads our image after converting it to grayscale and resizing it to 28 x 28 pixels. Lines 27 to 29 do

the necessary pre-processing so that the image can be given as input to our trained model. Line 30 predicts the class to which the image belongs to. Lines 31 to 33 print the details of this prediction. Figure 8 shows this part of the output of the program *digit.py*. From the figure it can be seen that, though the image is correctly identified as 7, the accuracy is only 23.77%. Further, from the value of the list score shown in Figure 8, it can be verified that the same number was identified as 1 with 12.86% accuracy and as 8 or 9 with about 11% accuracy. Moreover, the model even failed to identify the digit correctly on a few occasions. Though I couldn't pinpoint any reasons for this sub-par performance, I think the relatively low resolution (28 x 28 pixel) training images as well as the quality of the scanned images I have generated could be the main contributing factors. Though not the best model available in town, we now have our first trained model based on AI and machine learning principles. Hopefully, in the coming articles in this series, we will build models



Figure 7: A handwritten digit for testing

capable of performing even more difficult tasks.

Now, it is time to wind-up our discussion. We have started to learn about scikit-learn in this article, a discussion we will continue in the next article in this series also. Later,

```
Test loss: 0.022164013236761093
Test accuracy: 0.9940000176429749
1/1 [=====] - 0s 54ms/step
[[0.12864351 0.09797814 0.09591229 0.07170997 0.04486876 0.05580429
 0.0376949 0.23779215 0.1152301 0.11436587]]
Number is 7
Accuracy 23.779214918613434
```

Figure 8: Output of the Python script *digit.py*

we have seen more ways to deepen our understanding of neural networks. We also used Keras to train our first model and made predictions with this trained model. We will continue exploring neural networks and model training in the next article. We will also get familiar with PyTorch, a

machine learning framework based on the Torch library. PyTorch is used for developing applications involving computer vision and natural language processing.

Acknowledgement: I thank my student Sreyas S. for his creative suggestions during the preparation of this article. **END** 🐧

By: Deepu Benson

The author is currently working as assistant professor in Amal Jyothi College of Engineering, Kanjirappally, Kerala. He is a free software enthusiast and his area of interest is theoretical computer science.

www.opensourceforu.com

Open Source Is **Hot** In The IT World

Hadoop, Apache, Android, MySQL, Joomla, Drupal, PHP, OpenStack, PEARL, OSS

An EPY GROUP PUBLICATION

THE COMPLETE MAGAZINE ON **OPEN SOURCE**

OpenSource For You

Amazing How-To For SysAdmins

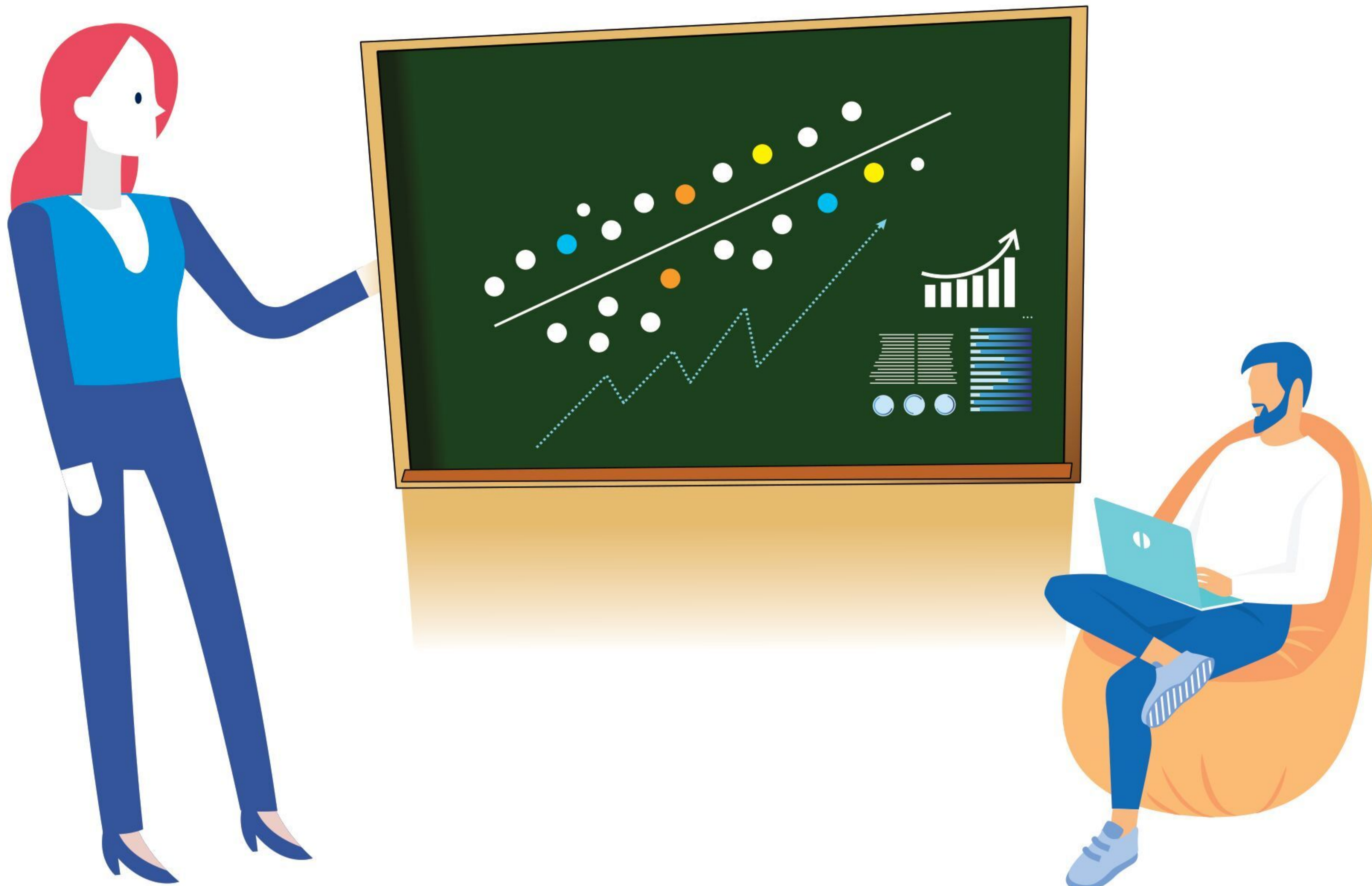
>> To find dealers to buy copy from news-stand, visit: www.ezine.ifymag.com

>> To subscribe, visit: pay.efyindia.com

>> To buy an ezine edition, visit: www.magzter.com choose Open Source For You

R Series: Regression

In this seventeenth article in the R series, we shall learn about linear regression in R.



In statistics, regression is a way to model the relationship between two variables. We will use R version 4.2.1 installed on Parabola GNU/Linux-libre (x86-64) for the code snippets.

```
$ R --version
```

```
R version 4.2.1 (2022-06-23) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
```

R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under the terms of the GNU General Public License versions 2 or 3. For more information about these matters see <https://www.gnu.org/licenses/>.

The mathematical equation for a linear regression between two variables 'x' and 'y' is defined as 'y = ax + b', where 'a' and 'b' are constants.

The *lm()* function is used to fit linear models. Consider two vectors 'x' and 'y' that represent odd and even numbers respectively. We can fit a regression equation between them using the *lm()* function, as follows:

```
> x <- c(1, 3, 5, 7, 9)
> y <- c(2, 4, 6, 8, 10)
> r <- lm(y~x)
> print(r)
```

```
Call:
lm(formula = y ~ x)
```

```
Coefficients:
(Intercept)          x
              1          1
```

The *lm()* function accepts the following arguments:

Argument	Description
formula	symbolic description of the model
data	data frame, list or environment
subset	optional vector of observations
weights	optional vector of weights
method	only 'qr' is supported
singular.ok	logical value, error if false
contrasts	optional list
offset	<i>a priori</i> known component
digits	significant digits

A more detailed output on the relation 'r' can be printed using the `summary()` function, as shown below:

```
> summary(r)
```

```
Call:
lm(formula = y ~ x)
```

```
Residuals:
    1      2      3      4      5 
4.367e-16 -8.276e-16  3.028e-16  1.303e-16 -4.222e-17
```

```
Coefficients:
            Estimate Std. Error  t value Pr(>|t|)
(Intercept) 1.000e+00  5.207e-16  1.920e+15 <2e-16 ***
x            1.000e+00  9.065e-17  1.103e+16 <2e-16 ***
---

```

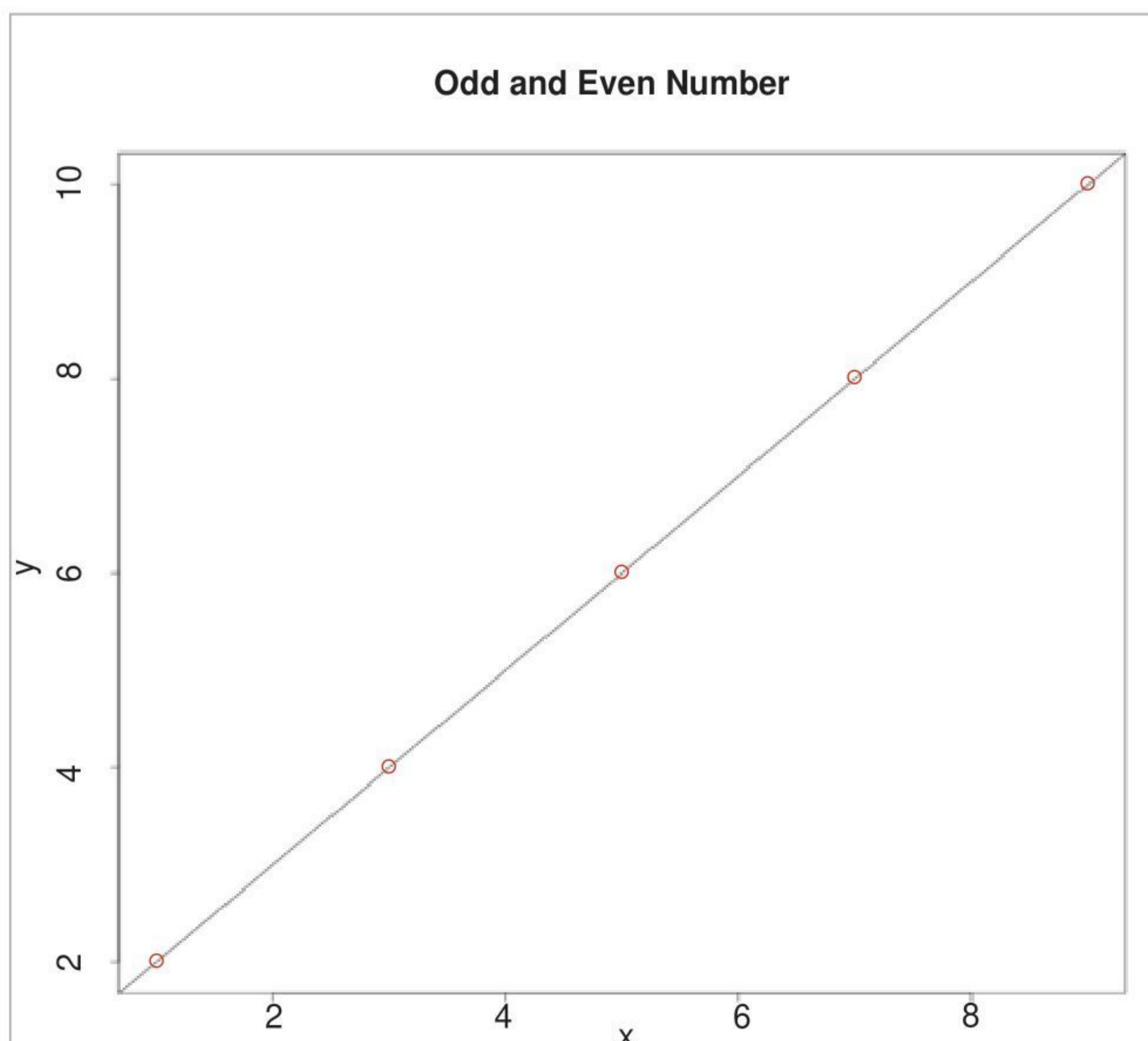


Figure 1: Plot x and y

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 5.733e-16 on 3 degrees of freedom
Multiple R-squared:  1, Adjusted R-squared:  1
F-statistic: 1.217e+32 on 1 and 3 DF, p-value: < 2.2e-16
```

We can use the `predict()` function to find the possible value of 'y' for a given 'x'. For example:

```
> v <- data.frame(x = 11)
> p <- predict(r, v)
> p
 1
12
```

The `plot()` function is used to display the fitted model, as illustrated below:

```
> plot(y, x, col="red", main = "Odd and Even Numbers",
       abline(lm(x~y)))
```

The normal QQ plot for the residual values can be displayed using the `qqnorm()` and `qqline()` functions, as shown below:

```
> qqnorm(r$resid)
> qqline(r$resid)
```

The kernel density estimates on the residuals can also be graphed using the `plot()` function, as shown in the figure.

```
> plot(density(r$resid))
```

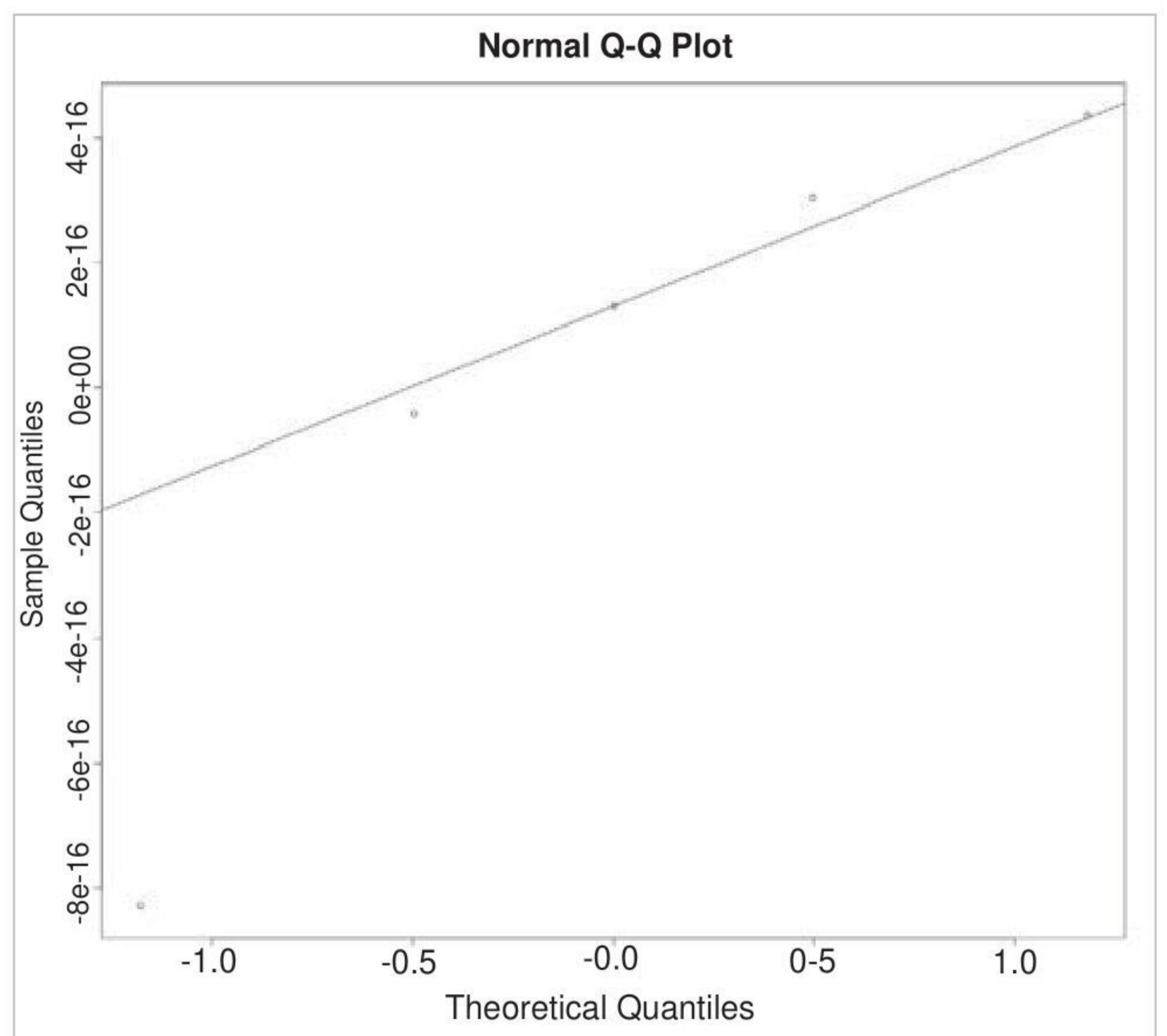


Figure 2: QQ plot

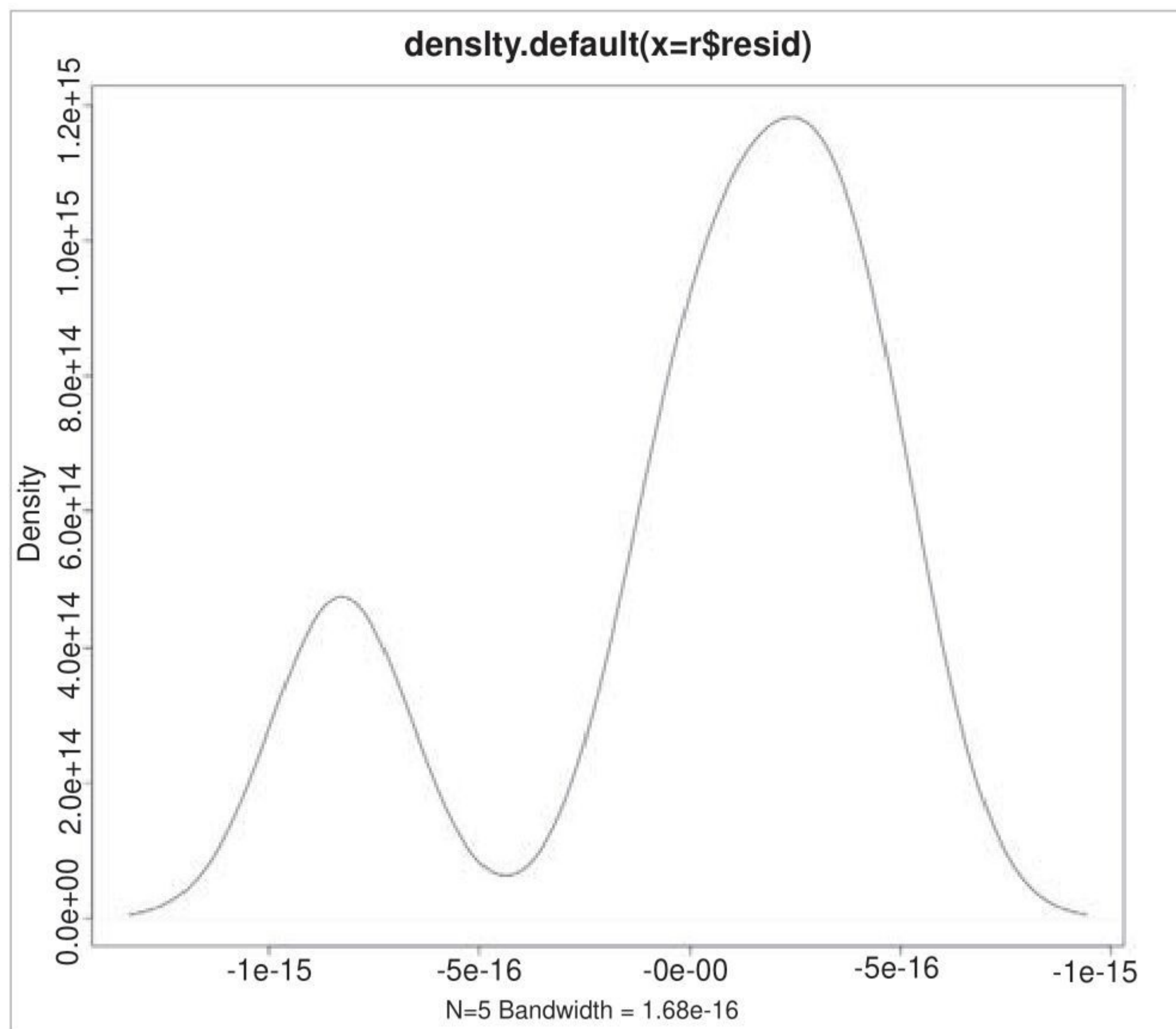


Figure 3: Density plot

The `ls()` function on the relation lists the various components of the 'r' object. For example:

```
> ls(r)
[1] "assign"    "call"     "coefficients" "df.residual"
[5] "effects"  "fitted.values" "model"       "qr"
[9] "rank"     "residuals" "terms"      "xlevels"
```

The components are briefly described below:

Component	Description
assign	integer vector for each column
effects	named numeric vector
rank	numeric rank of the model
call	the matched call
fitted.values	the fitted mean values
residuals	response - fitted values
coefficients	named vector of coefficients
model	model frame used
terms	the 'terms' object
df.residual	residual degrees of freedom
qr	QR decomposition
xlevels	levels of the factors used in fitting

The individual components can be viewed by prepending them with the '\$' on the relation, as illustrated below:

```
> r$coefficients
(Intercept)      x
              1      1

> r$assign
[1] 0 1

> r$effects
(Intercept)      x
-1.341641e+01  6.324555e+00  0.000000e+00 -4.440892e-16
-8.881784e-16

> r$rank
[1] 2

> r$call
lm(formula = y ~ x)

> r$fitted.values
 1  2  3  4  5
 2  4  6  8 10

> r$residuals
      1      2      3      4
5
4.367205e-16 -8.275904e-16  3.027995e-16  1.302899e-16
-4.221962e-17

> r$model
  y x
1  2 1
2  4 3
3  6 5
4  8 7
5 10 9

> r$terms
y ~ x
attr(,"variables")
list(y, x)
attr(,"factors")
  x
y 0
x 1
attr(,"term.labels")
[1] "x"
attr(,"order")
[1] 1
attr(,"intercept")
```

```
[1] 1
attr(,"response")
[1] 1
attr(,".Environment")
<environment: R_GlobalEnv>
attr(,"predvars")
list(y, x)
attr(,"dataClasses")
      y      x
"numeric" "numeric"

> r$df.residual
[1] 3

> r$qr
$qr
  (Intercept)      x
1 -2.2360680 -11.1803399
2  0.4472136  6.3245553
3  0.4472136 -0.1954395
4  0.4472136 -0.5116673
5  0.4472136 -0.8278950
attr(,"assign")
[1] 0 1

$qraux
[1] 1.447214 1.120788

$pivot
[1] 1 2

$tol
[1] 1e-07

$rank
[1] 2

attr(,"class")
[1] "qr"

> r$xlevels
named list()
```

The *confint()* function computes the confidence intervals for the parameters in the fitted model. For example:

```
> confint(r)
      2.5 % 97.5 %
(Intercept)  1      1
x            1      1
```

The *confint()* function accepts the following arguments:

Argument	Description
object	fitted model object
parm	vector of numbers or names for the parameters
level	confidence level required
...	additional arguments

The analysis of variance tables for the fitted model can be obtained using the *anova()* function, as follows:

```
> anova(r)
Analysis of Variance Table

Response: y
      Df Sum Sq Mean Sq    F value    Pr(>F)
x       1    40      40 1.2169e+32 < 2.2e-16 ***
Residuals 3     0     0
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Consider the *mtcars* data set available in the *lattice* library:

```
> head(mtcars)
      mpg cyl disp  hp drat   wt  qsec vs am gear carb
Mazda RX4  21.0  6  160 110 3.90 2.620 16.46 0  1  4  4
Mazda RX4 Wag  21.0  6  160 110 3.90 2.875 17.02 0  1  4  4
Datsun 710   22.8  4  108  93 3.85 2.320 18.61 1  1  4  1
Hornet 4 Drive  21.4  6  258 110 3.08 3.215 19.44 1  0  3  1
Hornet Sportabout 18.7  8  360 175 3.15 3.440 17.02 0  0  3  2
Valiant     18.1  6  225 105 2.76 3.460 20.22 1  0  3  1
```

We can fit a regression between the number of cylinders and horsepower using the *lm()* function, as follows:

```
> m <- lm(cyl~hp, data=mtcars)
> print(m)

Call:
lm(formula = cyl ~ hp, data = mtcars)

Coefficients:
(Intercept)      hp
  3.00680      0.02168
```

A summary of the fitted model is shown below:

```
> summary(m)

Call:
lm(formula = cyl ~ hp, data = mtcars)
```

```
Residuals:
  Min       1Q   Median       3Q      Max
-2.27078 -0.74879 -0.06417  0.63512  1.74067

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 3.006795   0.425485   7.067 7.41e-08 ***
hp          0.021684   0.002635   8.229 3.48e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.006 on 30 degrees of freedom
Multiple R-squared:  0.693, Adjusted R-squared:  0.6827
F-statistic: 67.71 on 1 and 30 DF, p-value: 3.478e-09
```

Some of the individual components from the fitted model are listed below for reference:

```
> m$residuals
 Mazda RX4      Mazda RX4 Wag      Datsun 710      Hornet 4 Drive
0.608014994    0.608014994    -1.023364771      0.608014994
Hornet Sportabout  Valiant      Duster 360      Merc 240D
1.198584681    0.716432710    -0.319263348    -0.351174929
...

> m$model
              cyl hp
Mazda RX4          6 110
Mazda RX4 Wag      6 110
Datsun 710          4  93
Hornet 4 Drive     6 110
Hornet Sportabout  8 175

> m$terms
cyl ~ hp
attr(,"variables")
list(cyl, hp)
attr(,"factors")
  hp
cyl 0
hp  1
attr(,"term.labels")
[1] "hp"
attr(,"order")
[1] 1
attr(,"intercept")
[1] 1
attr(,"response")
[1] 1
attr(,".Environment")
<environment: R_GlobalEnv>
```

```
attr(,"predvars")
list(cyl, hp)
attr(,"dataClasses")
  cyl      hp
"numeric" "numeric"

> m$qr
$qr
              (Intercept)      hp
Mazda RX4          -5.6568542 -829.78980772
Mazda RX4 Wag       0.1767767  381.74189579
Datsun 710          0.1767767   0.12620114
Hornet 4 Drive      0.1767767   0.08166844
Hornet Sportabout   0.1767767  -0.08860368
...


> confint(m)
              2.5 %      97.5 %
(Intercept) 2.13783849 3.87575200
hp          0.01630186 0.02706522

> anova(m)
Analysis of Variance Table

Response: cyl
              Df Sum Sq Mean Sq F value    Pr(>F)
hp              1  68.517   68.517   67.71 3.478e-09 ***
Residuals     30  30.358    1.012
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can again use the *predict()* function to determine the number of cylinders required for a given horsepower. In the following example, we observe that in order to achieve 200 horsepower, we require at least 7 cylinders.

```
> hp <- data.frame(hp = 200)
> cyl <- predict(m, hp)
> cyl
      1
7.343504
```

You are encouraged to read the manual pages for the above R functions to learn more on their arguments, options and usage. **END** 

 By: Shakthi Kannan

The author is a free software enthusiast.

Integrating Network Function Virtualization with the DevOps Pipeline: Distributed Systems

This multi-part series of articles will take you through the various components of DevOps. It will introduce you to network function virtualization and its integration with the DevOps pipeline. This first article in the series focuses on distributed systems.



Image Source: <https://www.freepik.com>

Service providers around the globe are looking for concrete solutions to provide the latest telecom applications and comply with the demands of their users. Proprietary networking devices are used to deploy these applications. While this solution fulfils customer requirements, it is not scalable and demands hefty financial investment because such hardware can easily get outdated with new technologies emerging in the market.

Network function virtualization (NFV) is seen as something that brings service providers out of this bottleneck. It is seen as a cost-effective implementation of telecommunications infrastructure. This series of articles will deal with the integration of NFV modules in a DevOps pipeline, while showcasing how various DevOps tools are interconnected to produce an environment capable of handling and deploying an NFV based infrastructure on an open source cloud.

With the advent of DevOps tools and methodologies, we now have a system for NFV applications that can be deployed and delivered in a jiffy. The various methodologies of DevOps fulfil the requirement of fast-paced delivery with easy-to-configure servers and no downtime of the application. NFV application, when orchestrated by the Kubernetes cluster, enables independent and dynamic scaling. NFV system enablement reduces the CAPEX and

OPEX too. This series of articles provides step-by-step guidance on how to integrate NFV with the DevOps process.

Since the experimental work involves many technologies, we plan to publish our work in eight parts that focus on distributed systems, containerization, Kubernetes, open source cloud computing, software-defined networking, NFV, DevOps methodologies, and NFV integration with DevOps.

The need today is for fast and powerful machines that can host our applications or store important documents, which should be available around the globe. This cannot be achieved via a monolithic system design. We need something that is robust and reliable, and available from our laptops, workstations, and mobile devices. This has led to a ‘distributed system’, first coined by Paul Baran, where the idea is to create a distributed communication network in which messages are sent through a network of switching nodes until they reach the desired destination.

The distributed system we see today is different from what was envisioned back then, but the basic concept remains the same. Such a system is basically a coalition of multiple machines that are viewed as one and come together to achieve a task.

Maarten van Steen and Andrew S. Tanenbaum in their book ‘Distributed Systems’, state: “A distributed system is a collection of autonomous computing elements that appears to its users as a single coherent system.”

This defines two important characteristics of such systems. The first is that these are collections of multiple computing elements, which can behave and work independently but are brought together. And the second is that they are viewed by users as a single unit, so the person using such a system does not differentiate between a single unit or multiple units. We can envision these systems as a graph (Figure 1) — each

computing unit is a node in that graph and each node is connected to another node via some network.

Architecture

By now we may look at distributed systems as a bunch of loosely interconnected processing devices working together over an underlying network, but there is more to this. The architectural styles define how these processing units are connected, and there

are several styles:

- Layered architectures
- Object-based architectures
- Resource-centered architectures
- Event-based architectures

Layered architectures: The idea behind layered architectures is to organise components of distributed systems in a layered manner. Here layer L_j makes requests to L_i where $(i < j)$, and expects a response in return (as shown in Figure 2).

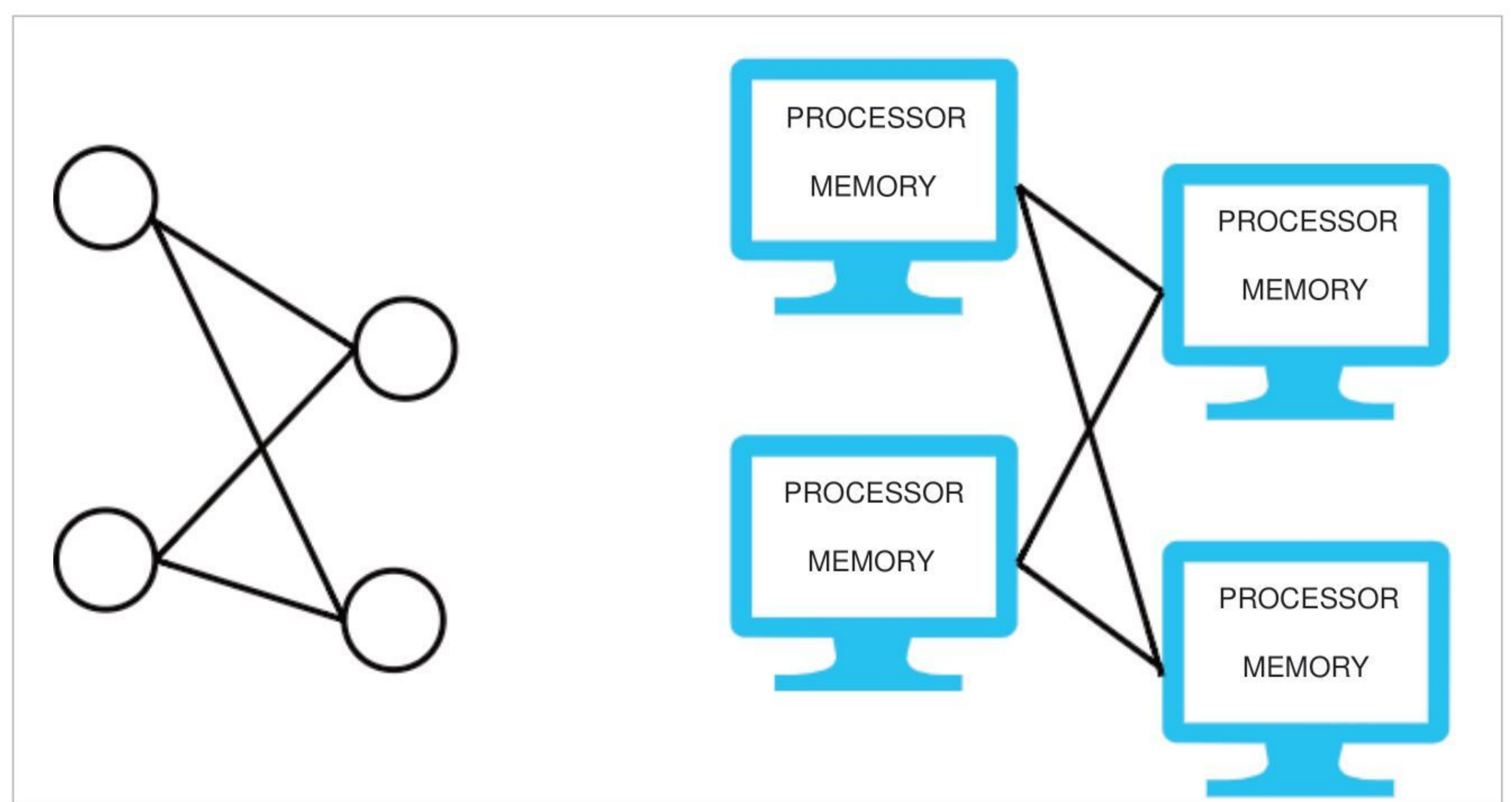


Figure 1: Node graph

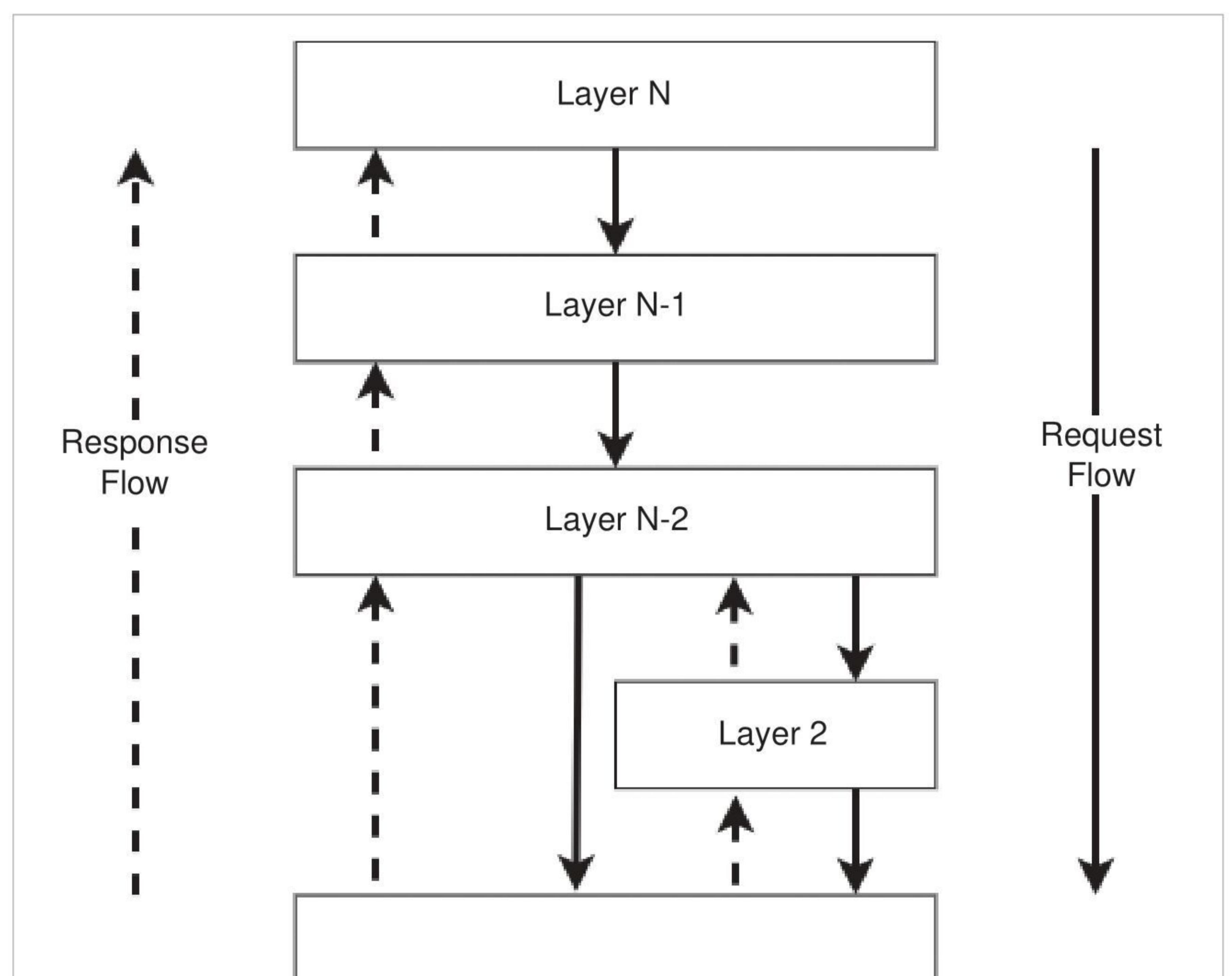


Figure 2: Layered organisation of various components

Object-based architectures:

Here we envisage each component in distributed systems as an object. In object-based architecture (Figure 3), we see one object calling another object via a procedure-calling mechanism. This can also happen over a network via a remote procedure call (RPC). We can separate services so they can operate independently, which is today known as service-oriented architecture (SOA).

Resource-centered architectures:

With the growing service composition to facilitate distributed systems, there was a need to rethink the aspect of web-based distributed systems; new architecture had to be defined to optimise the usage of distributed resources. Resource-centered architectures are robust mechanisms where we can add or remove the resources to be used by applications, based on their usage. Similar resources can be retrieved or modified. These characteristics of resource-centered architectures helped representational state transfer (REST) in web-centric applications gain wide acceptance. Various RESTful operations are showcased in Figure 4.

Event-based architectures:

In a large set of distributed systems, we try to decouple the dependencies between the processes. The idea behind this is to view processes running autonomously, and then there is the coordination between these processes which allows communication and data sharing between them. This helps adopt a separation between processing and coordination. In event-based architectures, as shown in Figure 5, after the separation is enabled, processes subscribe to the notification from the other components. These notifications are delivered to only those processes that have subscribed to them. This architecture uses an event bus to match the publisher and subscribers, and communicate the notifications.

System architecture

We have already looked into how the architectural styles are commonly applied in distributed systems. Now let us look at how software components

talk to one another, and how and where they are placed. Deciding on software components, their interaction, and their placement leads to an instance of software architecture, also known

as system architecture. Distributed systems can be categorised into centralised and decentralised architectures. In a centralised architecture, we consider simple server-client architecture, three-tier architecture, and n-tier architecture. When we talk about decentralised architecture, we consider peer-to-peer architecture.

Simple server-client architecture: Clients make a request to the server; so this request is to be served. The response to that request is served and any permanent changes are stored by the server.

Three-tier architecture: Here the tier denotes user interface, business logic, and data access.

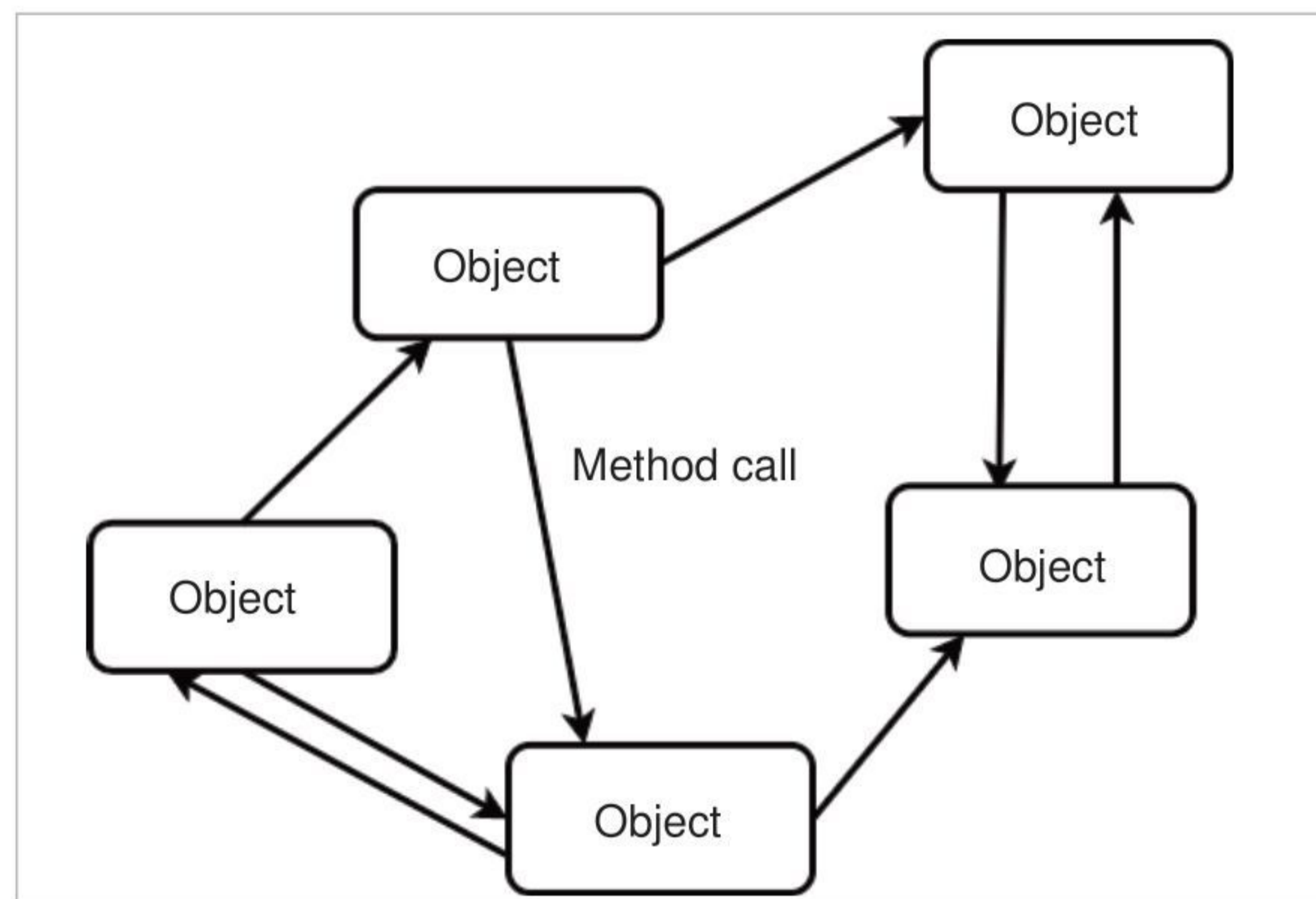


Figure 3: Object-based architecture with various components communicating with one another

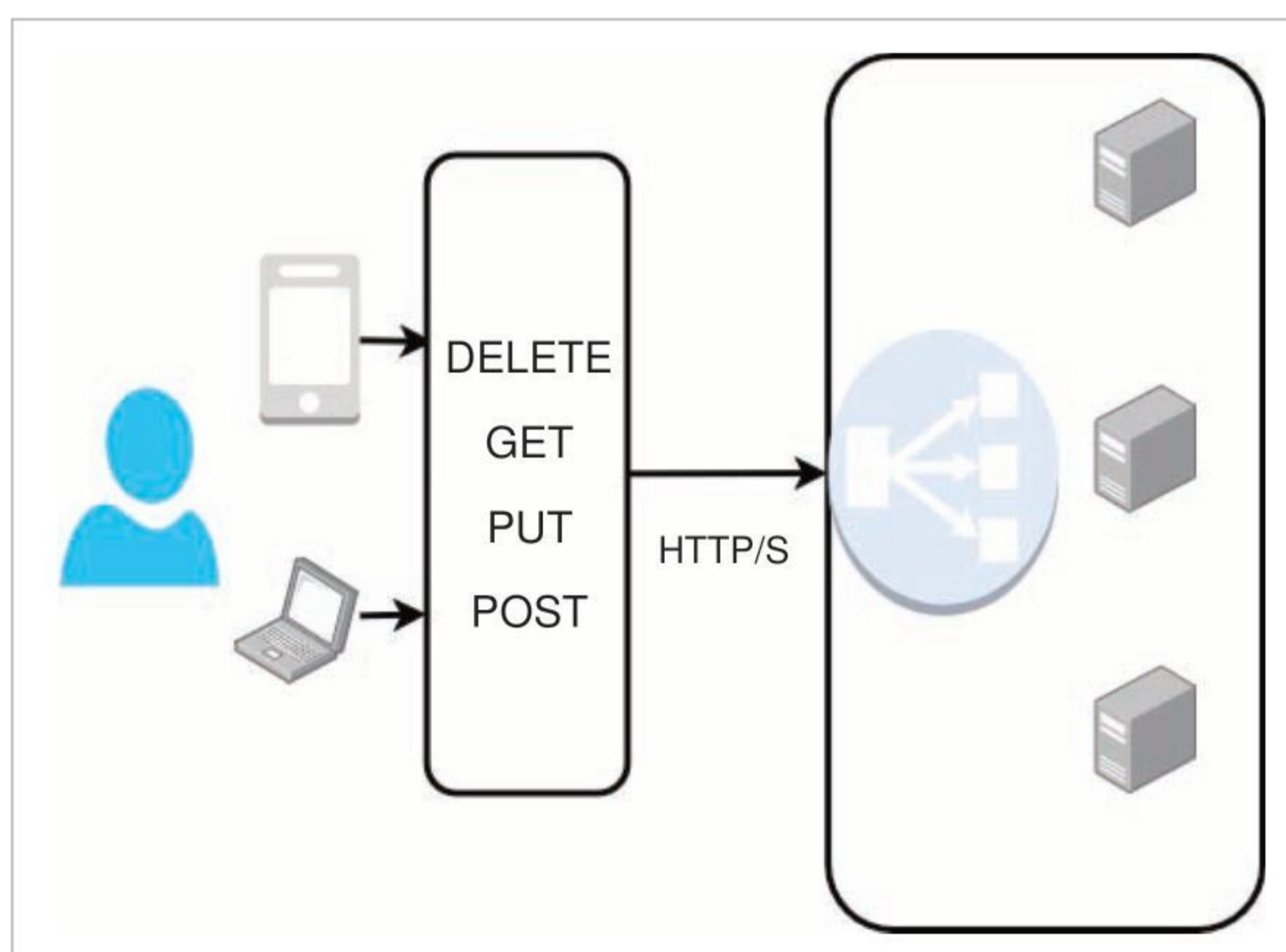


Figure 4: Resource-centred design with various RESTful operations

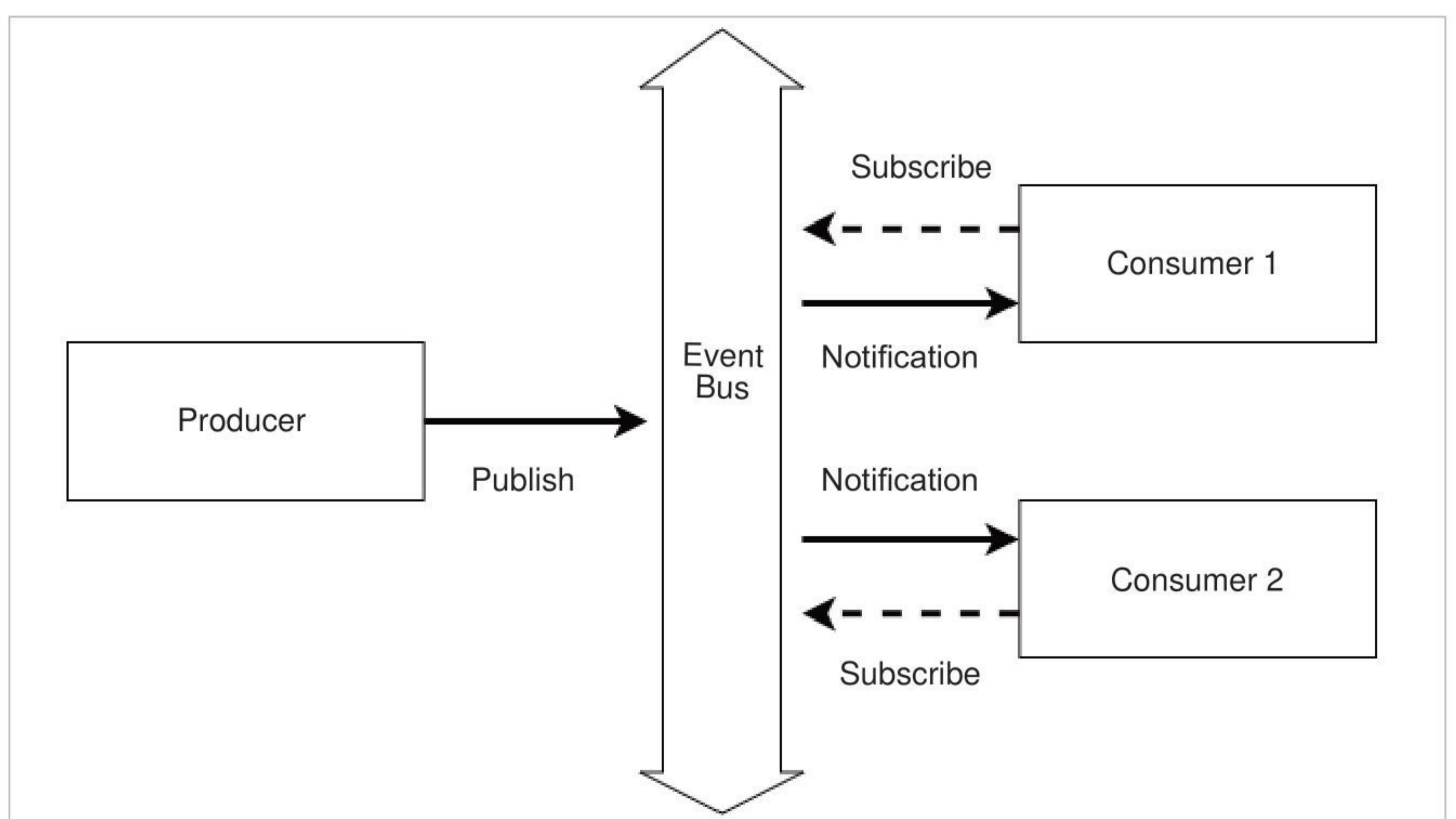


Figure 5: Multiple components in event-based architectural style

The collaboration of these three tiers is mainly used in web applications.

n-tier architecture: Other than the client, application server, and database server, the application server is further divided into multiple tiers, where each server acts as an entry point to do a different kind of processing on the provided data.

Peer-to-peer architecture: The peers are both servers and clients on a network. They work together to provide computing, storage, and network resources to network users. The responsibility for a task is uniformly distributed among the users of the network — these users are called peers.

Applications of distributed systems

Fault-tolerance: In case one of the nodes in the computer network goes down, to handle the current node, the load can be balanced to other subsequent nodes.

Multi-layered: In a distributed architecture, several computers come together over a network to do a specific task; for example, one computer node collects the data at location A, while the other processes it at location B.

Cost-effectiveness: Rather than investing in a single high-end computing device, the same computation can be achieved via clusters of several low-end devices.

Standard problems

In distributed systems, when we say every node is a computing unit, it must have an independent memory when it exchanges information by passing messages to other nodes. The first challenge that arises with this communication is how to do it efficiently so that the message is passed regardless of the network. Since computational problems are typically related to graphs, distributed algorithms are designed to solve them.

Distributed algorithms run on computation units, where each unit is

connected to another over a network. These algorithms are well known to solve some of the standard problems

in distributed systems, which include leader election, consensus, distributed search, spanning tree generation,

mutual exclusion, and resource allocation.

Let us take an example of a basic computation problem of distributed systems, where we see how a distributed system finds the colouring of a given graph G. As discussed earlier, we visualise the distributed systems model as a graph G, where G is a connection between computing units over a network. Every vertex or node in graph G denotes the computing unit. In finding the colouring of this graph, we see each node of G in a communication link with another node denoted via an edge. Initially, each node only knows about its immediate neighbours in the graph,

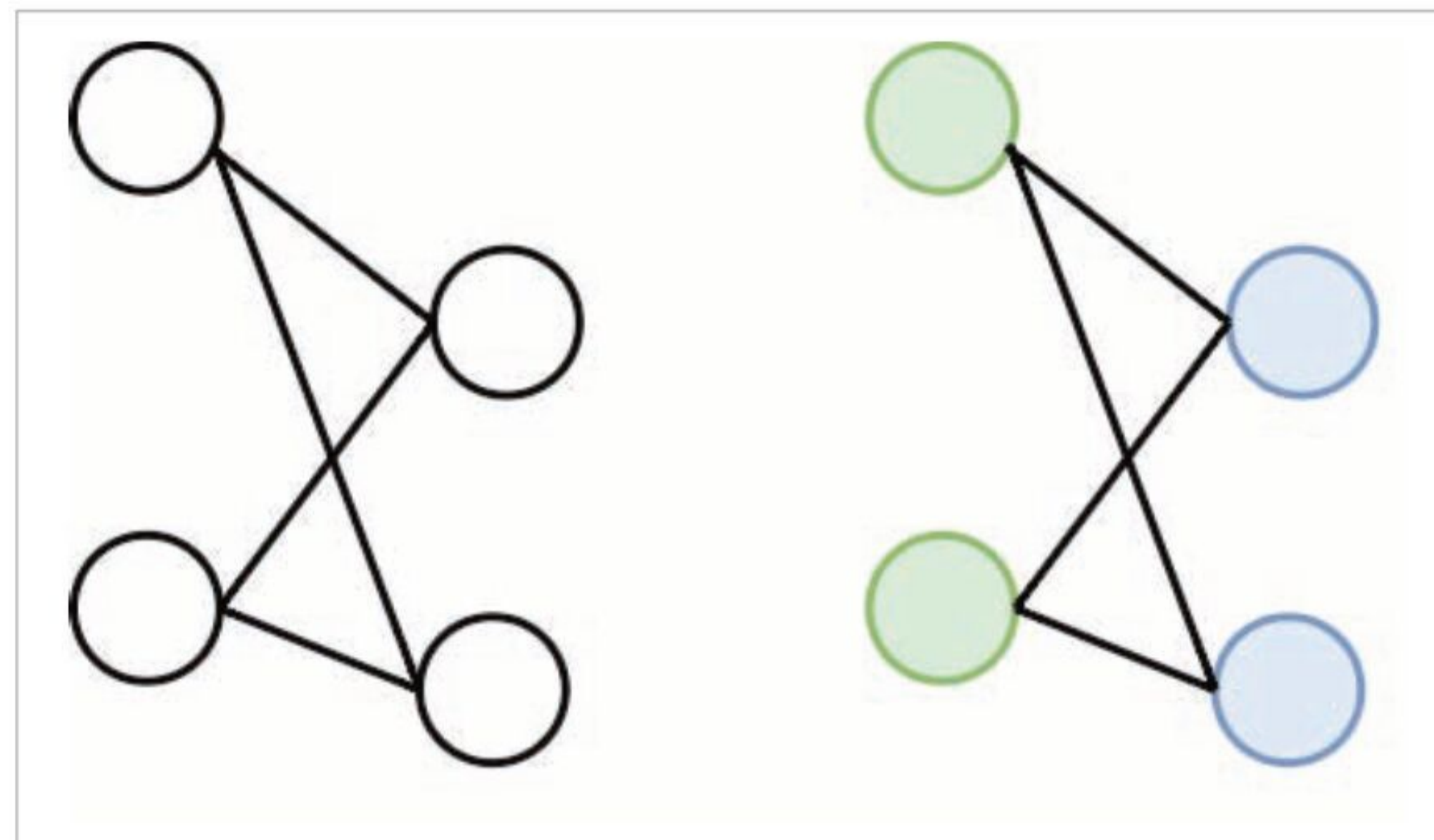


Figure 6: Colouring of a node graph connected in a network

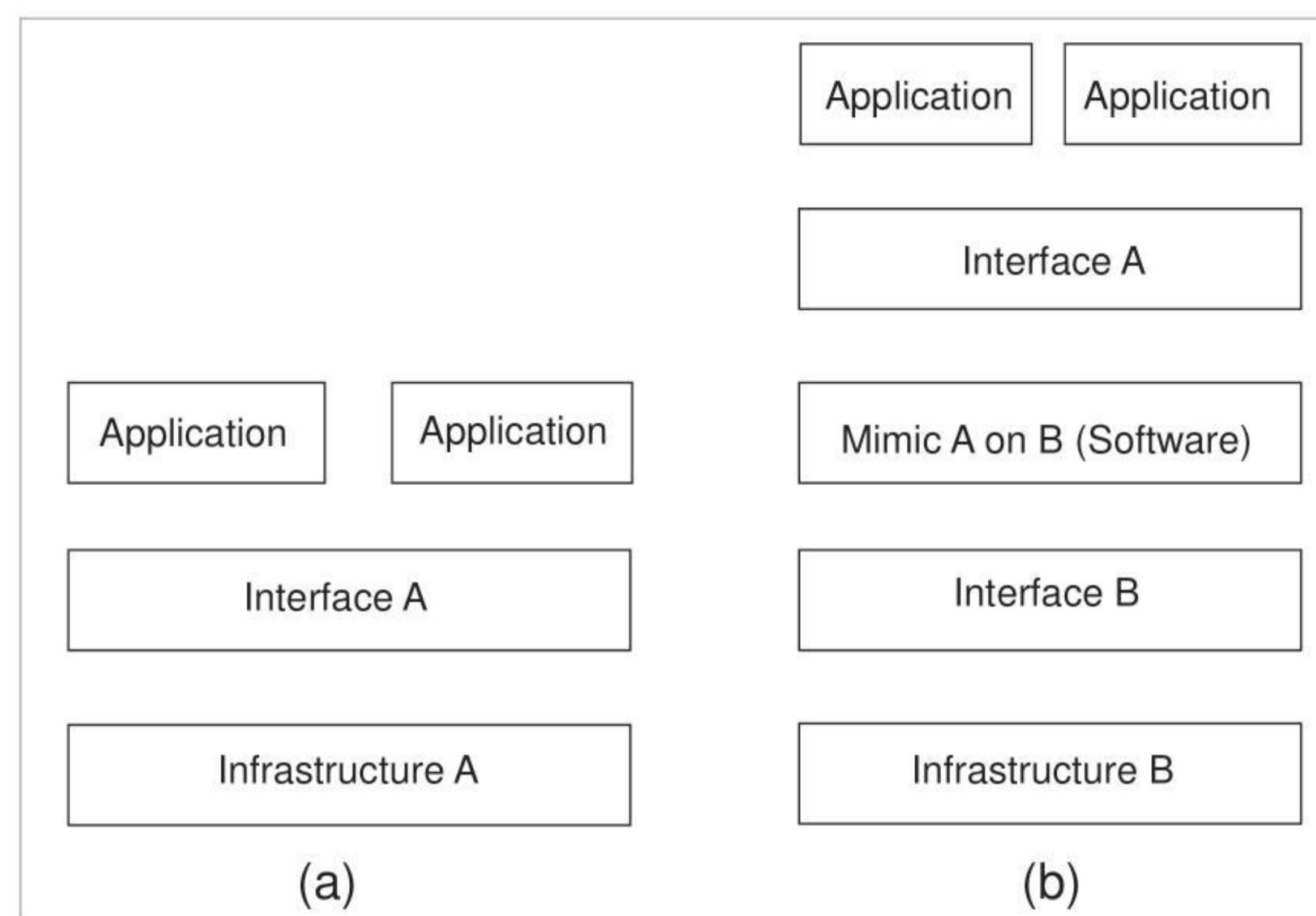


Figure 7: (a) Basic setup between infrastructure, interface, and application (b) Basic setup with virtualization of system A on system B

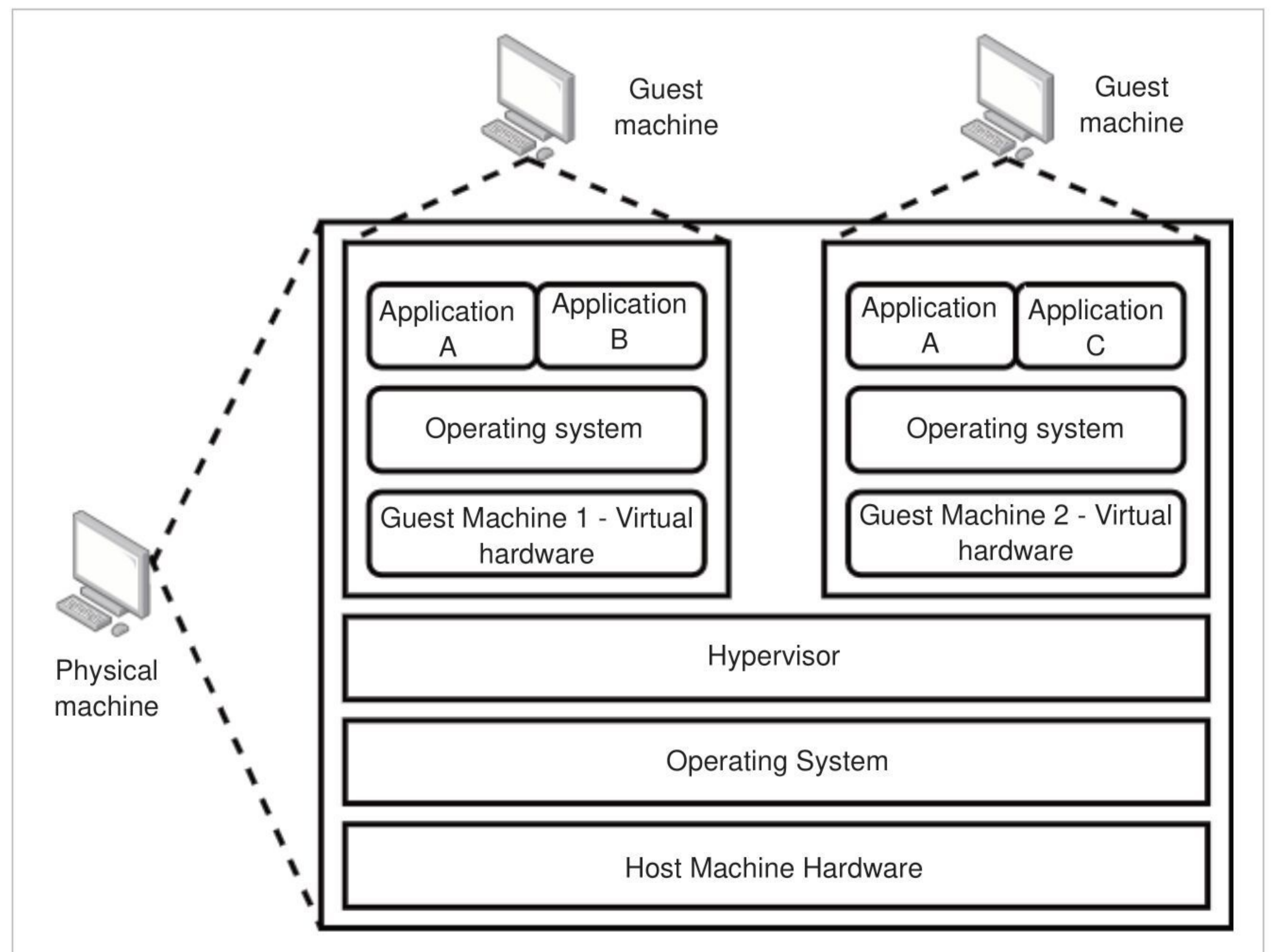


Figure 8: Hardware virtualization

and the nodes must exchange information with the entire network, keeping the information in sync while discovering more nodes in the given graph structure. With this approach, each node must produce its colour as output (as shown in Figure 6), effectively overcoming the barrier of distributed systems.

Networking

Over a distributed system, processing and storage resources are sitting apart from one another and only the network connects them. They can communicate with one another and share data via message passing. The communication processes follow a set of protocols to pass messages. Some of the widely used models of communication are:

- Hypertext transfer protocol
- Remote procedure call
- Message queues

Hypertext transfer protocol (HTTP): HTTP is widely used in web-based computing, where the client and server connect over the distributed network. The client sends a request to the server over a network using HTTP and expects a response. In multi-tier applications, the servers can use HTTP to communicate and pass messages to one another.

Remote procedure call (RPC): RPC is a form of inter-process communication between different processes running in the distributed system. The idea is to allow the calling of procedures located on different machines. RPC is request/response communication. A local instance of RPC runs on each computing node of the distributed systems, where if system A wants to run a remote procedure that is there on system B, A will ask the local instance of RPC to send the request to system B with a set of parameters, while RPC manages the transmission of parameters and other requests in the message over the network. After

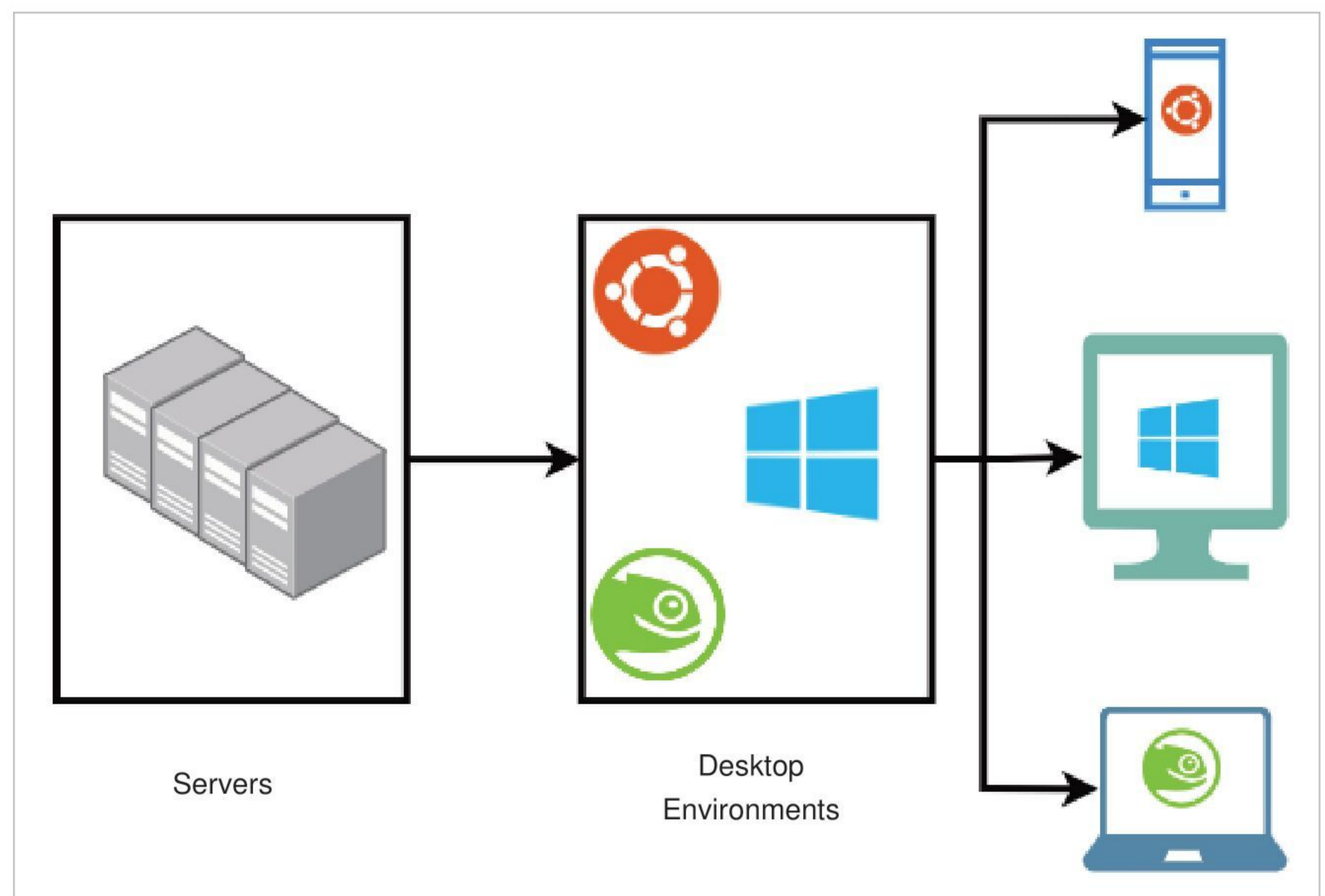


Figure 9: Desktop virtualization

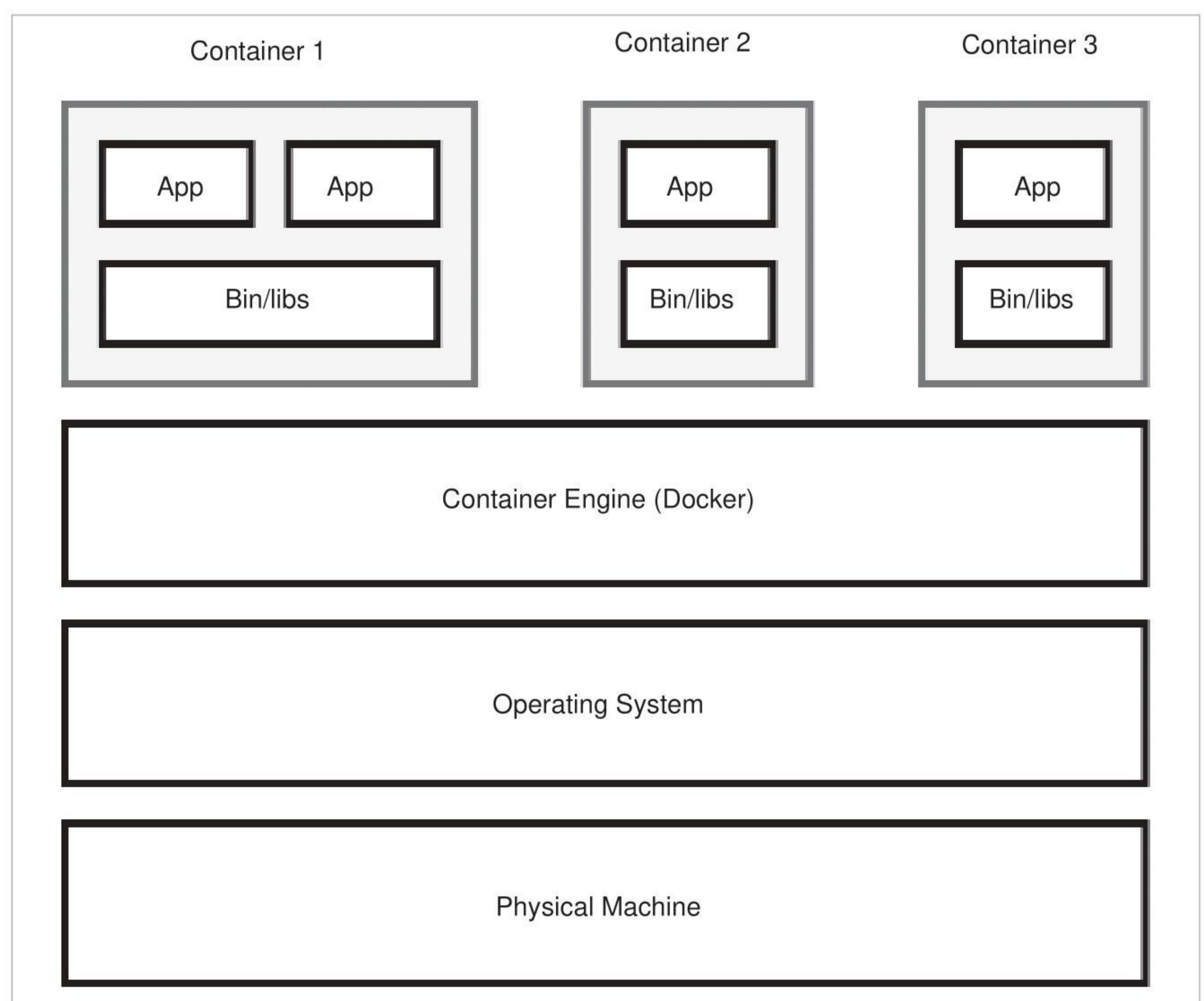


Figure 10: Containerization

the completion of the system, the B procedure is initiated by A, and the information is communicated to the callee from the caller (to A from B).

Message queues: In message queues, we use queues to communicate the message between distributed systems. The message

queue works on the asynchronous communication pattern, where even if the server is down when the message was communicated, it will still be received by the destined server when it comes back up. Each application has its private queue, where messages are delivered. They are stored on the

queue until the message is retrieved by the destined server. A few open source choices of message queues are RabbitMQ, Apache Kafka, and Apache ActiveMQ.

Virtualization

When we think of virtualization, we think of a methodology that separates our resources into different environments. The latter can have different execution environments based on the resources allocated by the users. These resources can be based on computing power, memory, disk space, and network. In virtualization, we are asking the existing interface to mimic the behaviour of another system.

Virtualization can help its users diversify the development and production environment. Based on the application, the user can decide on the operating system flavour and the libraries. Virtualization enables a high degree of flexibility and portability. Many tech giants like Amazon Web Services (AWS) and Azure offer virtual machines (VMs), and users can host their applications on these VMs. These modern-day distributed systems are called data centers and clouds. The infrastructure difference between the deployment of regular and virtualized applications can be seen in Figure 7.

Virtualization can be categorised into three types:

- Hardware virtualization (Type-1)
- Desktop virtualization (Type-2)
- Containerization (Type-3)

Hardware virtualization:

Suppose you want to run Ubuntu on your Windows machine — sounds tricky, right? Well, it's possible with hardware virtualization. In this sort of virtualization, we have a guest machine over the host machine. These terminologies — host and guest — are used to define a physical machine and a virtual machine, respectively. This

separation of the guest machine from the host machine is done via software or firmware, known as a hypervisor.


Desktop virtualization: For hardware virtualization, we were hosting a guest machine over the host machine. But what if we want to separate the logical desktop from the hardware itself? This is where desktop virtualization comes in. When users want to interact with such a desktop setup via desktop virtualization, they first connect to a network and then use a keyboard, mouse or touchpad to connect to it. The virtualized desktop can be accessed via another machine or mobile device. The host machine in this scenario can cater to multiple virtual machines at the same instance for multiple users. With desktop virtualization, an organisation can enjoy perks like scalability and a reduction in capital expenditure.

Containerization:

Containerization can be classified as OS level virtualization. It is a feature of the modern-day operating

system where the kernel allows the existence of multiple isolated user-space instances of these containers. All the requests by these containers are then forwarded to the host machine's kernel to entertain them. For the user the container may look like a real computer, but it is just another process for the host machine. The best thing about being isolated in the container is that any application or service running on it can only see the container's contents and its attached devices assigned by the host machine. A few important containers are:

- Docker
- AWS Fargate
- Canonical LXC
- OpenVZ
- CoreOS

In the next part of this series of articles, we will discuss containerization in detail, including container architecture, container engine, container networking, and container runtime. Till then keep exploring virtualization! **END** 

References

- Jonathan Effrat Natalie Ahn, Julie Black. Distributed computing, <https://cs.stanford.edu/people/eroberts/courses/soco/projects/2001-02/distributed-computing/html/history.html>. Accessed: 19-11-2021.
- Maarten Van Steen and Andrew S. Tanenbaum, 'Distributed Systems'. Maarten van Steen Leiden, The Netherlands, 2017.
- Bruce Jay Nelson. 'Remote Procedure Call'. Carnegie Mellon University, 1981.
- SOA sourcebook - what is SOA? <https://collaboration.opengroup.org/projects/soa/book/pages.php?action=show&ggid=1314>. Accessed: 28-11-2021.
- Roy Thomas Fielding. Fielding dissertation: Chapter 5: Representational state transfer (rest). http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.html, 2000.
- Len Bass, Paul Clements, and Rick Kazman. 'Software Architecture in Practice'. Addison-Wesley Professional, 2003.
- Gregory R. Andrews. 'Foundations of Multithreaded, Parallel, and Distributed Programming', volume 11. Addison-Wesley Reading, 2000.
- Gerard Tel. 'Introduction to Distributed Algorithms'. Cambridge University Press, 2000.
- Nancy A Lynch. 'Distributed algorithms'. Elsevier, 1996.
- 'An Introduction to Virtualization'. https://webx.ubi.pt/~hgil/utills/Virtualization_Introduction.html. Accessed: 1-12-2021.
- Bernard Golden. 'Virtualization for dummies'. John Wiley & Sons, 2011.

 **By: Shubham Aggarwal, Nithya Ganesan, and Dr B. Thangaraju**

The authors are associated with the Open Source Technology Lab in the International Institute of Information Technology, Bengaluru.

Thank You IIT Madras

It was in your hostel room
that the idea was conceived



The first issue in Jan '69



The Jan '22 Issue

Celebrating 54 years in TECH media

► Magazines

- Electronics For You
- Electronics Bazaar
- Open Source For You

► Portals

- www.electronicsforu.com
- www.opensourceforu.com
- www.eleb2b.com
- www.electronicb2b.com

► Events

- India Electronics Week
- IOTshow.in
- EFY Expo
- Test & Measurement India
- Open Source India
- Techworldcongress.com

“

**I started reading it
when I was a student...
...and I am still
reading it, as a student**”

—CEO, Design House



electronics
YOURS SINCE 1969 **FOR YOU**



To Subscribe:
<https://subscribe.efy.in>

OR

Scan This Code



For any query, call: +91-98111-55335 or email: support@efy.in