Mastering Immutable Infrastructure: Step-By-Step Guide To Building Golden AMI

The Challenges Of Using AI Tools To Generate Code

₹120
ISSN-2456-4885

OpenSource For You

Volume: 11 | Issue: 11 | Pages: 100 | September 2023

THE COMPLETE MAGAZINE ON OPEN SOURCE

An EFY GROUP Publication

DevOps

Enabling The Success Of DevSecOps With AI

How DevOps Differs From Traditional IT, And Which One Is Better

DevOps As A Career: Education, Experience and Opportunities



Coq And Lean: Powerful Interactive Theorem Provers

Streamlining Software Delivery With Tekton Pipelines

www.opensourceforu.com

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.

India's #1 Event For

R&D and New Product Development

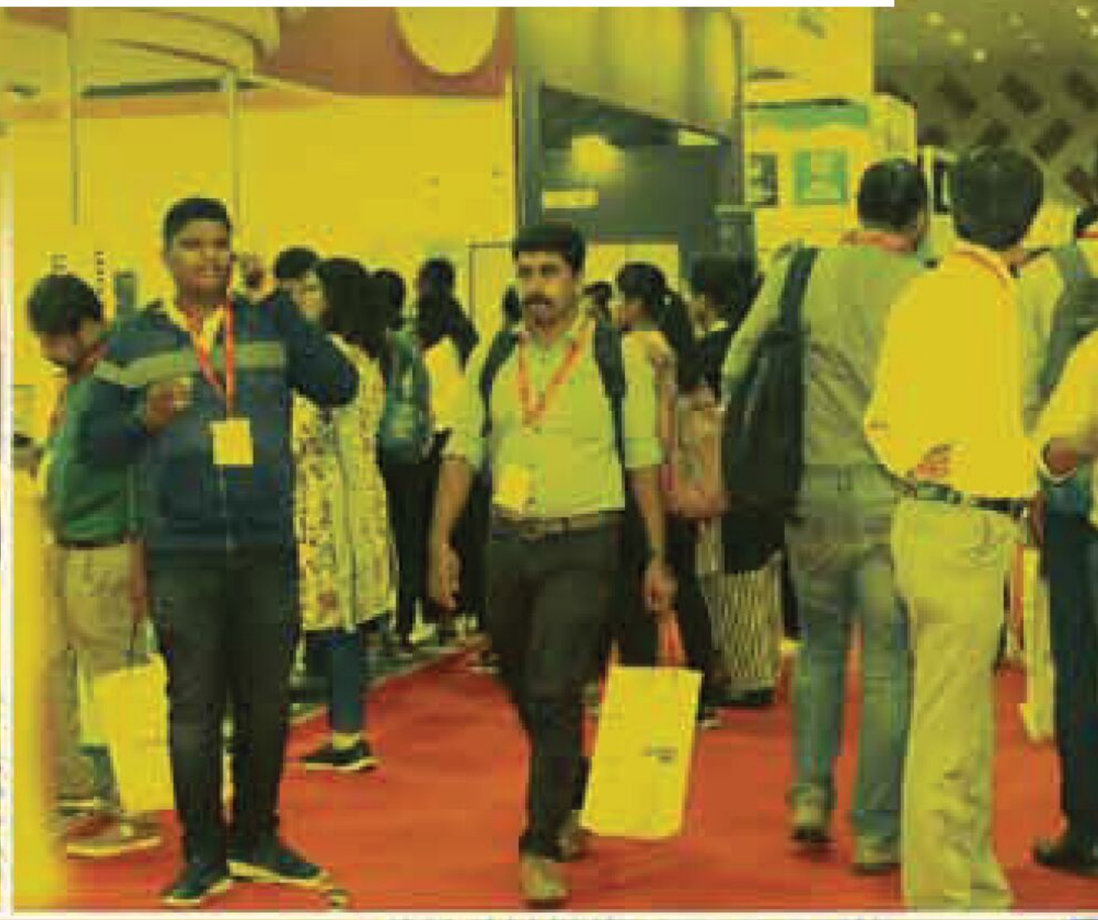
Of Electronics

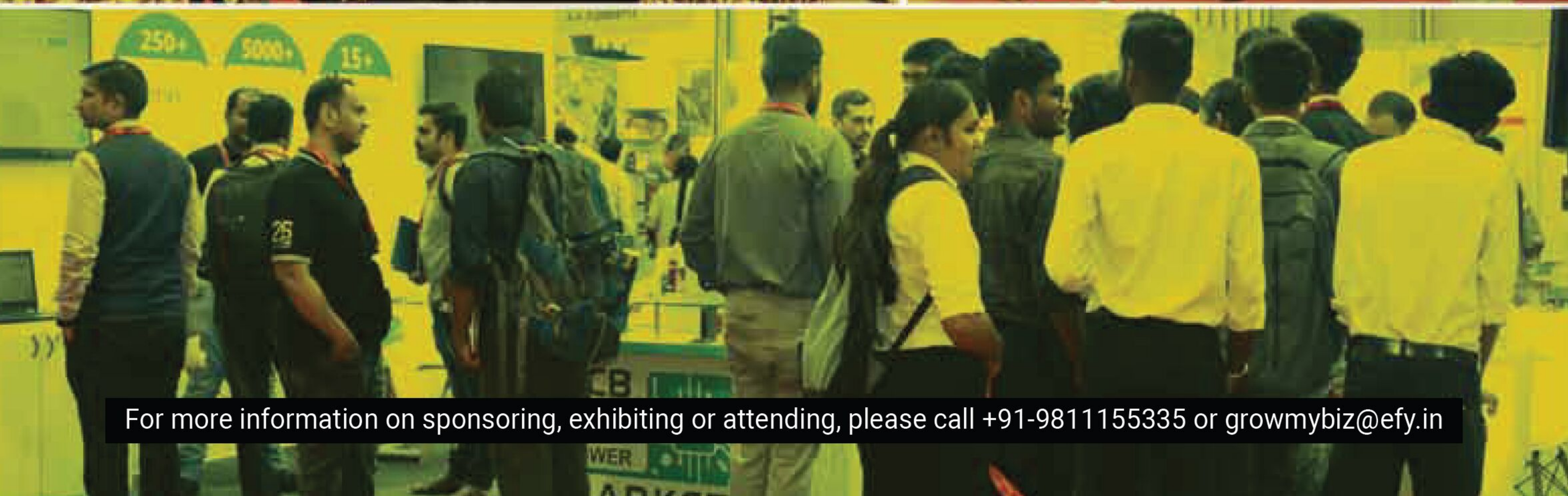
9TH EDITION INDIA ELECTRONICS WEEK 2024

31st Jan - 2nd Feb 2024

KTPO Expo Center |
Whitefield | Bengaluru | India







For more information on sponsoring, exhibiting or attending, please call +91-9811155335 or growmybiz@efy.in

CONTENTS

SEPTEMBER 2023 | ISSN-2456-4885

FOR U & ME

12 The Challenges of Using AI Tools to Generate Code

FOCUS

21 Streamlining Software Delivery with Tekton Pipelines

29 MLOps: Benefits, Applications and Popular Platforms

33 Enabling the Success of DevSecOps with AI

38 Embracing Progressive Delivery in Kubernetes with GitOps

47 Continuous Integration with GitHub Actions

52 How DevOps Differs from Traditional IT, and Which One is Better

56 DevOps as a Career: Education, Experience and Opportunities

DEVELOPERS

59 R Series: mlr3 Package

63 Coq and Lean: Powerful Interactive Theorem Provers

73 Data Analysis: Web Scripting with Python

78 Building Fast and Feature-Rich Web Applications Using HyperApp

83 Demystifying Connection Pools: A Deep Dive

89 Unleashing the Power of ReactPy

ADMIN

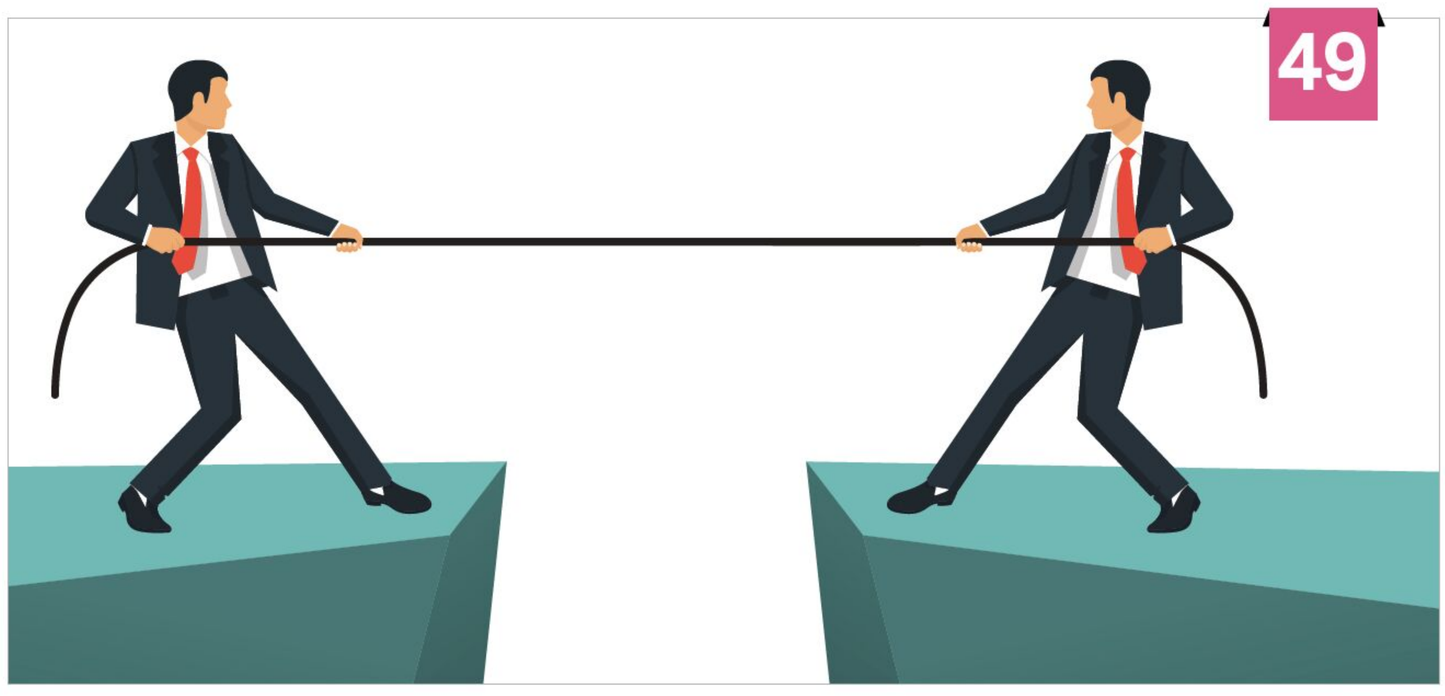
93 Mastering Immutable Infrastructure: A Step-by-Step Guide to Building Golden AMI



Data Analysis with PandasAI, the Generative AI Python Library



The AI and Cybersecurity Capabilities of GitLab's CI/CD Platform



DevOps Vs Agile: What You Should Know

REGULAR FEATURES

07 FossBytes

EFYGROUP
YOURS SINCE 1969

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

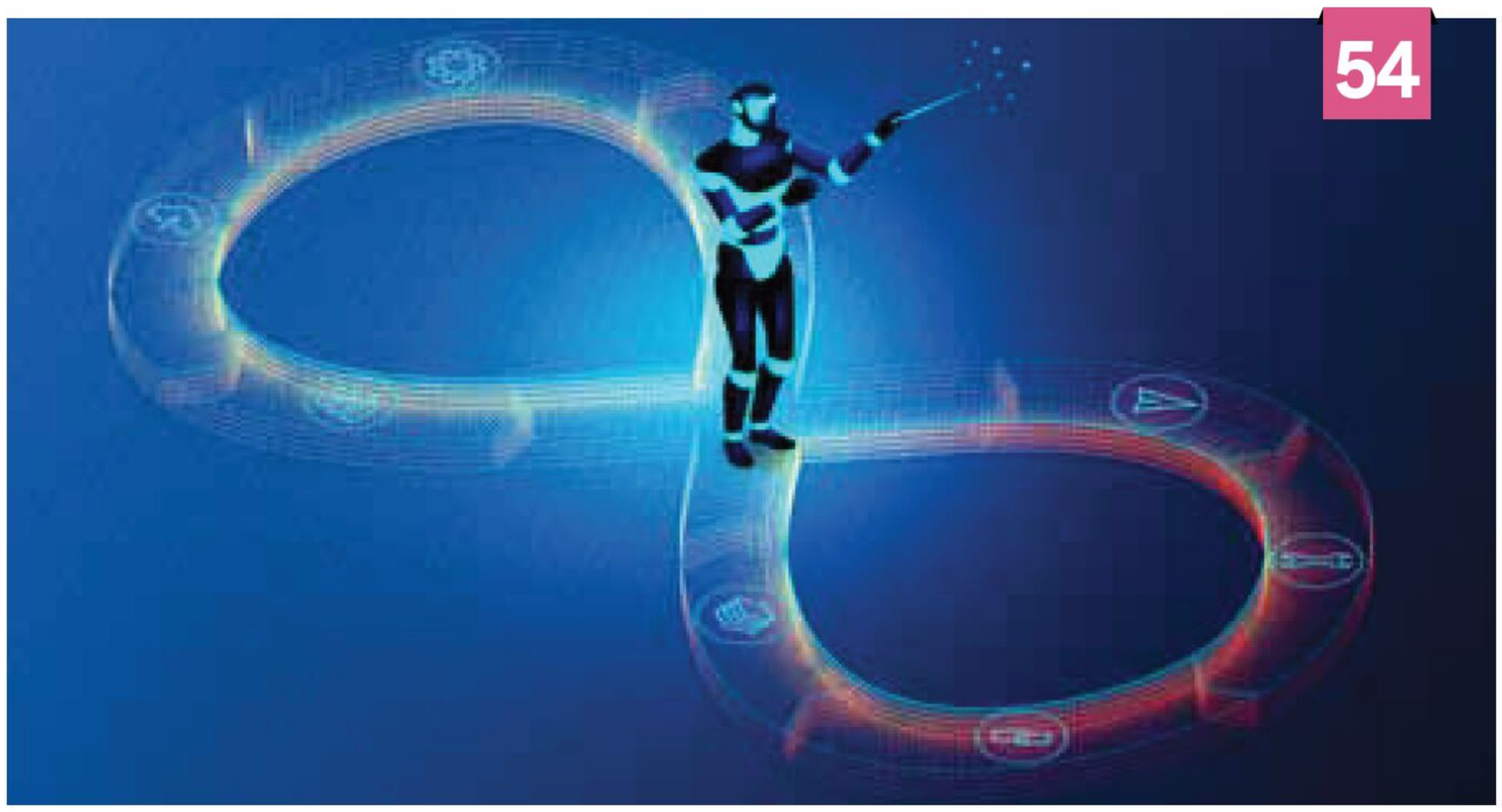
UNITED STATES
E & Tech Media
E-mail: usa@efy.in

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 © 2023. 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.

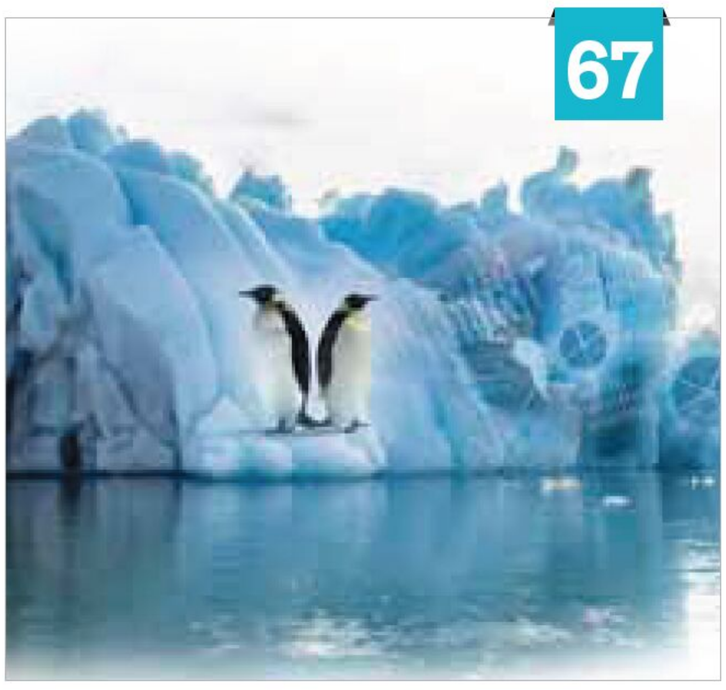
SUBSCRIPTION RATES			
Year	**Newstand Price (₹)**	**You Pay (₹)**	**Overseas**
Five	7200	4320	—
Three	4320	3030	—
One	1440	1150	US$ 120

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.

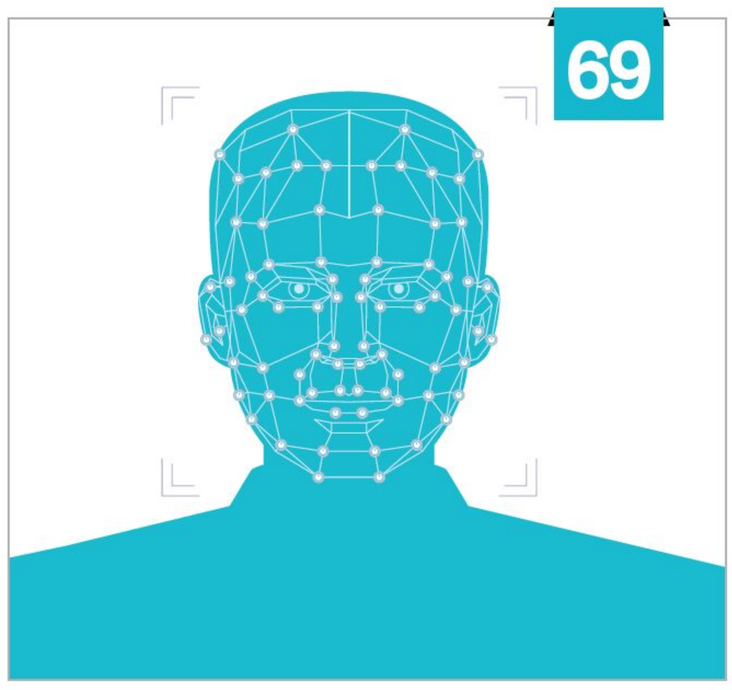
CONTENTS



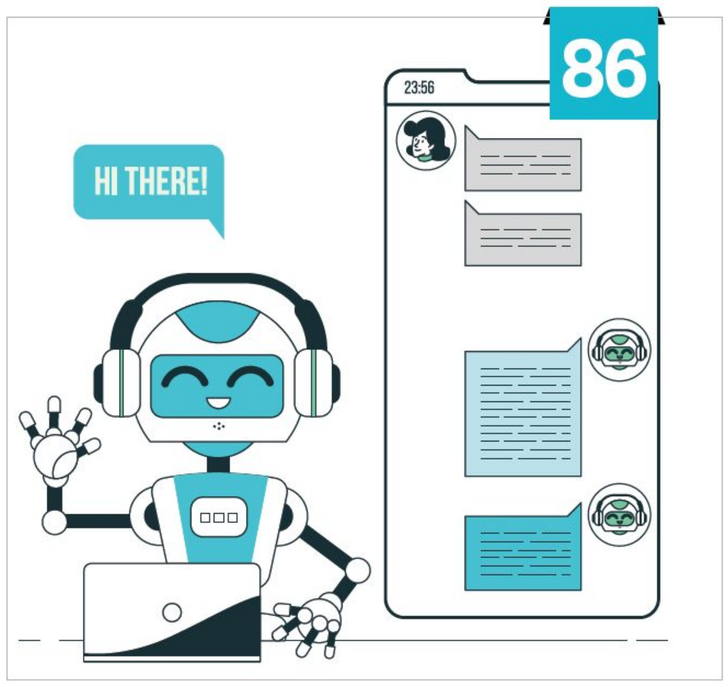
How AI/ML can Automate DevOps



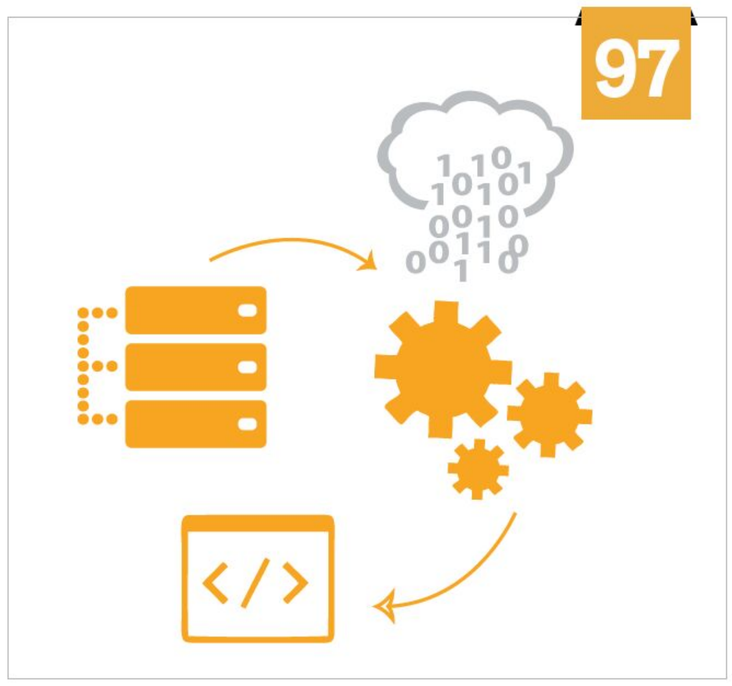
Using Apache Iceberg for Developing Modern Data Tables



OpenCV: Bringing AI Within Your Reach



Building a Chatbot Application with IBM Watson



Cloud Orchestration with Infrastructure as Code (IaC): The Benefits

Moq under fire for endangering data privacy by including SponsorLink



The widely used open source project Moq is embroiled in controversy after it was revealed that its latest release included a dependency known as SponsorLink. Moq, pronounced 'Mock', is a prominent player in the open source software ecosystem, boasting over 476 million lifetime downloads and a staggering 100,000 daily downloads from the NuGet software registry. The criticism erupted when Moq's 4.20.0 release quietly incorporated the SponsorLink project, prompting backlash from open source software enthusiasts who felt betrayed by what they deemed a breach of trust.

The root of the criticism lies in SponsorLink's deceptive nature—it is shipped on NuGet as closed source software, containing obfuscated DLLs (dynamic link libraries) that surreptitiously gather hashed email addresses of users and transmit them to SponsorLink's content delivery network (CDN). A prominent software developer, Georg Dangl, expressed his concerns, stating, "This is a closed-source project, provided as a DLL with obfuscated code, which seems to at least scan local data (git config?) and sends the hashed email of the current developer to a cloud service." This scanning capability is integrated into the .NET analyser tool during the build process, raising significant privacy concerns.

Mike (@d0pare), a GitHub user, decompiled the DLLs and shared a reconstructed version of the source code, revealing that SponsorLink leverages external git processes to obtain email addresses and then computes SHA-256 hashes of these addresses, which are subsequently sent to SponsorLink's CDN. The inclusion of such telemetry code within Moq and SponsorLink has raised questions about privacy and transparency within the open source community. Concerns have also been raised about the ethics of distributing a closed source dependency via open source channels, potentially compromising user data privacy.

In response to the backlash, Daniel Cazzulino, Moq's owner, updated SponsorLink's README with a detailed 'Privacy Considerations' section, asserting that only hashed email addresses were collected and that the actual

Open source propels software strategy at Porsche

Luxury car maker Porsche has ventured further into the realm of open source software. Porsche's recently published guide, titled 'FOSS Movement', articulates the brand's growing emphasis on leveraging free and open source software (FOSS) within its software strategy. Roughly two years after the inception of its open source initiative, the company is now making strides by sharing its fundamental principles through this publication.

The document elucidates the core values underpinning free and open source software and outlines the roles that employees, spanning departments such as vehicle development, corporate IT, legal, and Porsche Digital, play in collaborating with these technologies. There is a commitment to uniformly embed open source practices across Porsche's operations. This integrated approach not only enhances internal cohesion but also fosters external collaboration with open source communities on a global scale.

The influence of open source permeates Porsche's digital transformation, underpinning crucial aspects of its software ecosystem. From the intricate control units within vehicles to pivotal elements like the 'My Porsche' digital platform and the 'Porsche Design System', the power of open source is evident.

Porsche's collaboration with open source projects also serves as a conduit for discovering and nurturing fresh talent. This engagement allows the company to interact closely with a global cadre of technological leaders, fostering an environment of cross-pollination and shared expertise.

SandboxAQ introduces Sandwich, a cryptography framework

SandboxAQ's latest offering 'Sandwich' introduces a transformative open source framework set to redefine modern cryptography management. Developers can now seamlessly integrate cryptographic algorithms into applications, adapting to evolving technologies and security threats without rewriting code. The framework's unified API simplifies this process while providing enhanced control for improved cybersecurity.

"Modern cryptography management and cryptographic agility are essential, yet open source tools supporting these features have been lacking," noted Graham Steel, head of product for SandboxAQ's Quantum Security Group.

Sandwich is highly compatible with the libOQS cryptographic library, offering easy access to cutting-edge post-quantum cryptography (PQC) algorithms from NIST. This is pivotal in fortifying digital defence against quantum threats.

Moreover, Sandwich facilitates customisation. Developers can build tailored 'sandwiches' of protocols and implementations, avoiding disruptive code recompilation during reconfiguration. This high-level API also aids in steering clear of common cryptographic pitfalls.

"We created Sandwich to accelerate development of our encryption management and cryptographic remediation solutions," Steel explained. This altruistic approach empowers developers to experiment with agile cryptography, enhancing community preparedness against quantum encryption breaches.

email addresses were never transmitted. Despite the rollback of the controversial change in Moq's version 4.20.2, concerns linger about the possibility of similar occurrences in future releases.

France bets on open source AI to shape AI industry

French President Emmanuel Macron recently unveiled a groundbreaking initiative aimed at bolstering the nation's presence in the AI realm. However, this aspiration is not without its challenges, as the European Union's AI regulations loom large.



The plan centres on nurturing open source AI, which involves making computer code publicly available for anyone to use, modify, and enhance. The crux of this strategy lies in circumventing the hurdles posed by the EU's stringent AI regulations. These regulations are designed to ensure responsible AI development and usage but can also hinder innovation due to their intricate compliance requirements.

By championing open source AI, France aims to leverage a potential loophole in the EU's upcoming Artificial Intelligence Act, which could exempt open source systems from certain regulatory constraints. French President Macron's announcement at the prestigious VivaTech conference underscores the nation's commitment. He pledged a significant €40 million investment to establish an open "digital commons" for French-made generative AI projects, and declared, "We believe in open source."

Macron announced a substantial €500 million investment to cultivate AI 'champions' that can lead the market and research efforts in this emerging field. Notably, one of these champions is the Paris-based startup Mistral.ai, which raised €105 million in funding to create an open source alternative to OpenAI's ChatGPT.

However, industry leaders remain cautious, questioning the viability of the EU's regulatory exemptions. Open source AI systems are only exempted from strict compliance rules if they are not used for commercial purposes. This limitation could potentially deter investors and startups from fully embracing open source AI.

Indian Defence Ministry to replace MS Windows with the home-grown Maya OS

The Indian Defence Ministry has announced its decision to replace the widely-used Microsoft Windows operating system with an indigenously built alternative, named Maya. Developed locally over a span of six months by government agencies such as the Centre for Development of Advanced Computing (C-DAC), the Defence Research and Development Organisation (DRDO), and the National Informatics Centre (NIC), Maya OS is based on the open source Ubuntu Linux distribution.



Maya's USP is its compatibility with commonly used applications like Microsoft Office, Adobe Photoshop, and AutoCAD, ensuring a smooth transition for users. This feature-rich operating system also incorporates advanced attributes such as cloud storage, encryption, digital signatures, and biometric authentication, all of which contribute to heightened security and functionality.

Kiran Vangaveti, founder and CEO of BluSapphire Cyber Systems, highlighted the key advantage of Maya OS: "One valuable aspect is the increased control

over the operating system's functionality. This facilitates adaptability for custom applications, industrial automation, and seamless functioning of defence software."

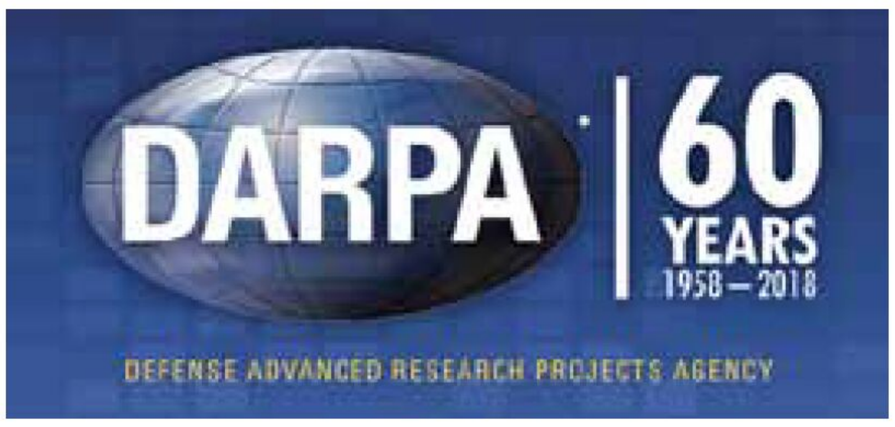
Contrary to popular belief, Maya OS isn't India's maiden attempt at an indigenously developed operating system. In 2007, C-DAC introduced the Bharat Operating System Solutions (BOSS) with the goal of promoting free and open source software adoption across India. BOSS, available in 19 Indian languages, was adopted by the Indian Army too.

The shift to Maya OS is primarily driven by the imperative to fend off cyberattacks and malware threats. Notably, the Indian Army faced a barrage of cyberattacks, with a report from 2019 indicating at least two attacks every month. Maya OS is poised to enhance security, cyber resilience, and diminish dependency on external systems. The heightened control over data and sensitive information is another boon that Maya OS promises to offer.

Biden Administration kicks off 'AI Cyber Challenge' to boost open source software safety

The White House has unveiled an ambitious initiative that aims to leverage the power of artificial intelligence (AI) in identifying and rectifying vulnerabilities within open source software. This groundbreaking effort, known as the 'AI Cyber Challenge', was announced by the Defense Advanced Research Projects Agency (DARPA) in August, signalling a new chapter in the battle against cyber threats.

The competition, set to be led by DARPA, carries significant incentives for participants. Teams entering the fray will vie for prizes totalling a staggering $18.5 million. Additionally, the agency has allocated an extra $7 million in prize money specifically for small businesses that choose to partake in the challenge.



Central to the competition's objectives is the utilisation of AI technology to address software vulnerabilities, with a notable emphasis on open source software. The Biden administration has garnered support from prominent AI industry players, including Anthropic, Google, Microsoft, and OpenAI, all of which have pledged to make their cutting-edge AI technology available for the challenge.

This strategic move by the White House and DARPA comes in response to escalating concerns surrounding cyber supply-chain risk, both within the federal government and the private sector. The initiative also aligns with the recent commitments made by seven leading AI companies in July, as they signed onto a framework of voluntary pledges brokered by the Biden administration.

The AI Cyber Challenge will proceed with a structured timeline. Participating teams will undergo a qualifying event in Spring 2024, with the top 20 scoring teams advancing to the semi-finals at DEFCON 2024. From these, up to five teams will receive monetary prizes and move on to the final phase of the competition, scheduled to be held at the annual DEFCON Hacker convention in 2025.

Nephio R1 launched to simplify cloud-native automation for telcos

The Nephio project, announced by the Linux Foundation over a year ago, is making headway with its maiden major release, Nephio R1. This collaborative endeavour aims to simplify cloud-native automation for telecommunication companies.

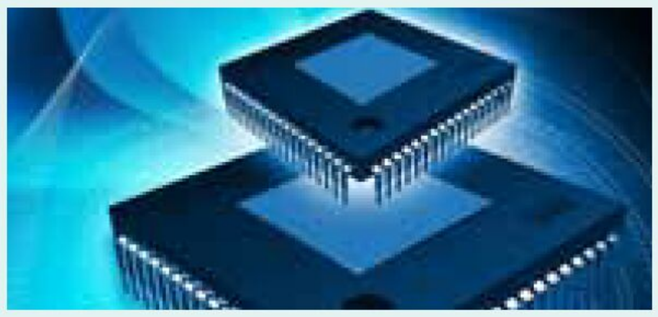
The project leverages Kubernetes as a foundational layer for the seamless management of cloud infrastructure and network functions, catering to the evolving demands of 5G operations.

India to back open source RISC-V architecture

India has thrown its support behind the open source RISC-V architecture. This move comes as RISC-V gains momentum as a potent contender on the global stage, challenging the dominance of Softbank's ARM technology.

Under the banner of the Digital India RISC V (DIR-V) programme, India's computing systems are set to harness the power of this open source architecture. The country's ambitions span various sectors including automotive, space technology, IoT sensors, and mobility, as highlighted by Rajeev Chandrasekhar, the minister of state for electronics and information technology.

Speaking about the initiative, Chandrasekhar affirmed, "The government is committed to making DIR-V the Indian ISA. There were some doubts if we will go the ARM way, but I want to commit to you today that DIR-V will have a central place in the high performance computing systems being developed by our country." The minister delivered this announcement virtually at the Digital India RISC-V Symposium, jointly organised by the Ministry of Electronics and Information Technology (MeitY) and IIT-Madras in Chennai.



The adoption of the open source RISC-V architecture not only signals India's aspirations for self-reliance but also secures its place at the forefront of global chip design developments.

As India takes its stand alongside the growing RISC-V community, it is clear that the nation is ready to make its mark in the semiconductor landscape.

Altrios from SwRI aims to make the rail freight industry greener

The Southwest Research Institute (SwRI) has launched open source software Altrios, which offers a practical approach to assess green locomotive technologies and infrastructure expansion.

Altrios, a product of collaboration between SwRI, the National Renewable Energy Laboratory, University of Texas at Austin, University of Illinois Urbana-Champaign RailTec, and BNSF Railway, combines locomotive and energy storage tech models with dispatch simulations and routing tools. It scrutinises decarbonisation strategies – like hydrogen, biofuels, and batteries – on real rail infrastructure, aiding informed choices. Altrios facilitates cost-effective evaluation of decarbonisation techniques through computer modelling.

As the US endeavours to decarbonise the transport sector by 2050, Altrios arrives at an opportune moment. Available for public download, Altrios is funded by the US Department of Energy. This software exemplifies the industry's collective efforts to reshape its environmental impact.

In an era of climate consciousness, Altrios will empower rail companies to embrace sustainability without compromising efficiency.



Kandan Kathirvel, chair of the Nephio Technical Steering Committee, said that Nephio R1 focuses on automating cloud infrastructure and network functions using Kubernetes. He underlined the importance of aligning network functions with global infrastructure, a significant move given the shift towards container-based deployment in recent years.

Nephio R1 adopts the Kubernetes Resource Model (KRM) approach for service deployment, replacing the customary Helm Charts. The GitOps-inspired KRM model introduces controlled version management and updates for smoother deployments.

Kathirvel highlighted Nephio's intent-based automation model, stating, "The user has to express the intent for what they really need and then the automation system is capable of identifying what configurations are needed."

Deployed as a separate Kubernetes cluster, Nephio serves as an automation engine for telcos, simplifying workload and cluster automation. Its versatility extends to network functions down to physical switches, with potential for customised Kubernetes operators. The project aims to bolster out-of-the-box automation through additional operators, advancing operational efficiency.

HashiCorp abandons open source for 'source available' approach

HashiCorp has announced a shift from its conventional open source stance to a 'source available' model. This is the company that was behind developing vital software tools such as Vagrant and Terraform. This transition follows the footsteps of MongoDB, Elastic, and Confluent, which have also pivoted to similar strategies, citing concerns about misuse and competition.

Co-founder Arman Dadgar, in a recent blog post, explained that HashiCorp's initial commitment to open source, which prioritised ecosystem building and user code adaptability, faced challenges due to some entities exploiting open source without contributing back. This prompted HashiCorp's adoption of the Business Source License (BSL) version 1.1 for future releases, effective from August 10th.

While APIs, SDKs, and various libraries will remain under the Mozilla Public License v2.0 (MPL 2.0), HashiCorp's move to BSL 1.1 means that users can still copy, modify, redistribute, and use the software under specific conditions. However, competitive offerings directly rivalling HashiCorp's products will be restricted.

Avi Press, CEO of Scarf, praised HashiCorp's handling of the transition. He said on Twitter/X, "HashiCorp has set a good bar for how to do a BSL switch smoothly. No misnaming anything, no attacks, just a difficult business decision carefully communicated. They are a well-meaning group of people who have shown they do care about OSS, whether or not you like their decision."

However, Peter Zaitsev, co-founder of Percona, noted the ongoing polarisation within the open source landscape. He described this move as "hostile towards the community who supported the company along the way."

This shift underscores the intricate interplay between community contributions, commercial interests, and competition within the open source domain. As HashiCorp joins the ranks of companies redefining their approaches, the broader open source ecosystem braces for potential impacts.

ONCD calls for public input on ways to strengthen OSS security

The Office of the National Cyber Director (ONCD) in the US has announced a call for public input on a strategy to fortify open source software (OSS) security. Teaming up with key players in the cybersecurity and technology fields, the ONCD's request for information (RFI) seeks to harness collective expertise in order to bolster the resilience and safety of open source software systems.



Scheduled to close on October 9, the RFI, a collaborative effort involving the Cybersecurity and Infrastructure Security Agency (CISA), the National Science Foundation (NSF), the Defense Advanced Research Projects Agency (DARPA), and the Office of Management and Budget (OMB), marks a significant step towards addressing the mounting security concerns surrounding open source software.

The impetus behind the RFI derives from the commitment outlined in the American National Cybersecurity Strategy. This strategy, a White House initiative, emphasises the need "to invest in the development of secure software, including memory-safe languages and software development techniques, frameworks, and testing tools."

Highlighting the broader implications of securing open source software, the RFI underscores its pivotal role in national security, economics, and technological innovation. As open source software finds itself deeply embedded across federal agencies and critical infrastructure, vulnerabilities within these components can propagate widespread and potentially damaging effects.

With a clear awareness of the benefits and vulnerabilities of open source software, the RFI envisions a concerted effort to elevate open source software to a national public priority.

Through the current RFI, ONCD, CISA, NSF, DARPA, and OMB aspire to pinpoint areas that deserve government attention, as well as to address critical queries such as mitigating systemic risks, ensuring the sustainability of open source communities, and devising technical and resource-based solutions for OSS security.

Türkiye unveils OSS initiative for tech advancement and cybersecurity

Türkiye has unveiled an ambitious open source software initiative aimed at revolutionising its public institutions. This strategic step, initiated by a decree signed by President Recep Tayyip Erdoğan, underscores the nation's commitment to technological advancement and cybersecurity.



The initiative's primary objective is to tap into the potential of open source solutions, fostering cost savings in IT expenditures, reducing dependence on external software vendors, and fortifying cybersecurity measures. The move aligns with a global trend recognising the pivotal role of open source software in promoting innovation and ensuring digital sovereignty.

Under this new roadmap, all public institutions are expected to integrate open source software. Failure to comply will require institutions to provide explanations, highlighting the gravity of the initiative. To facilitate this transition, a template has been established by the Presidency's Digital Transformation Office, requiring relevant institutions to submit their transition plans within nine months.

The software sector's representatives, backed by the Software Council of the Union of Chambers and Commodity Exchanges of Türkiye (TOBB), have been advocating for the adoption of open source software. This move not only opens significant market opportunities for domestic software companies but also eliminates the burden of chronic software licensing payments within the public sector.

Türkiye aims to prioritise open source alternatives over commercially licensed software for goods and services procurement, except in cases of technical or economic infeasibility. This approach aligns with the nation's commitment to self-reliance and security.

The move mirrors efforts by other nations to safeguard their interests in the digital realm. Notably, the United States has launched its open source software initiative to counter the influence of Chinese companies in the telecommunications sector.

For more news, visit *www.opensourceforu.com*

The Challenges of Using AI Tools to Generate Code



Enterprises must understand the obligations that underline OSS licences and adopt proactive measures to meet them while using AI-generated code. Otherwise, they may end up in legal entanglements.

Artificial intelligence, is a technique used to train machines to work and behave like humans. It encompasses various training methods and software like machine learning, neural network and deep learning, natural language processing, and more.

As the field of artificial intelligence (AI) advances, generative AI (GAI) techniques are gaining enormous attention for their ability to generate code and provide suggestions based on prompts that are provided by developers, thereby accelerating the software development process.

These AI models are typically trained on billions of lines of code, most of which are open and freely available to the public. Based on the developers' prompts, these AI tools generate a fully functional code or code snippets.

With the rise in the use of AI tools and developers utilising them to generate creative outputs, GAI will fundamentally change the way that software is built. Whether they are developing or using AI tools, organisations must understand the opportunities and risks involved, and evolve policies, processes, and governance to address them. This includes open source software codes.

A glimpse of open source software licences

Open source software (OSS) has become a major pillar of modern software development, fostering collaboration and innovation. With the increase in the availability of generative AI models, it has become essential to address the complexities of the code governed by open source licences.

The OSS licence allows users to access, use, modify, distribute, or redistribute the open source software. However, many OSS licences impose different obligations and conditions on the use and distribution of the software. These can range from providing attribution to specifying that the OSS modified code must be released under the same licence.

Failure to comply with these obligations can result in IP infringement, which may result in litigation and the loss of right to use the open source software.

OSS compliance issues with AI generated code

The legal concerns arising from AI-based code generator tools are caused by the processing of publicly available OSS code to train the AI

PURESTORAGE®

A New Era of Data Storage

Pure Storage® FlashBlade//E™ delivers the benefits of all-flash with better economics than disk.

Discover FlashBlade//E™

FlashBlade //E™



From Container to Multi-Cloud

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.

Slack GitHub SODA CDM

Want to know more and contribute?
http://bit.ly/soda-starter

KAHU

SODA Framework Projects

STRATO

Container Data Protection

- Container Data Backup/Restore
- Kubernetes Native Design
- Easy to add new Storage Providers
- CSI snapshot, NFS, OpenEBS providers & counting

Join us in developing data mover, replication and more.
https://github.com/soda-cdm/kahu

Multicloud Data Management

- Manage your data across multiple cloud vendors
- Unified interface for object, file and block
- Migration, Backup, Lifecycle, Storage Plans and more
- S3 Compatible API for hybrid object data

Join us in developing Metadata Management today.
https://github.com/sodafoundation/multi-cloud

www.sodafoundation.io

models. These codes are sometimes covered by certain licences.

To avoid the possibility of a lawsuit, some organisations have taken the approach of restricting their development teams' use of generated code. However, such restrictions will prevent developers from obtaining the benefits of AI code generators.

AI models used for code generation are very dynamic and continuously evolving. This might be a challenge in ensuring compliance with OSS licences, as the model behaviours keep changing, leading to new compliance issues. There are other challenges too.

- Training data may consist of a vast amount of OSS, including OSS libraries or snippets and sources that are incomplete or lack proper licensing information. It may contain software licensed under certain OSS licences, making it challenging to ensure these licences are compatible with each other.
- AI-generated code can contain snippets or patterns from open source projects. Identifying the licence for this generated code can be difficult, making it challenging to determine the compliance and associated obligations.
- Developers may wonder whether they are required to licence new software applications created from AI-generated code under the same open source licence.

Possible solutions

In light of these complexities, let's explore the practical remedies to mitigate legal risks.

- Just using open source code in the training model is unlikely to constitute any infringement. However, the obligations must be considered based on the licences under which the open source code is made available. If the obligations are not satisfied as per the licence, it can create IP risk.
- Use tools that have trained their models with reputable sources. Training the models on larger sets of data is expensive and requires significant computing power, which larger companies typically possess.
- Avoid relying on the output generated by the tools and using it as it is. Instead, it is always better to revise the output, use it as a reference and edit the generated code as per the requirements. This approach will help increase the chances of the output to be protected under copyright.
- It is always a good practice to segregate the AI generated code from the code not generated by AI tools.
- Periodically scan the project using software composition analysis tools (e.g., Black Duck, Code Insight) to identify the open source software libraries or the snippets that are under restrictive licences in the project.
- Many AI-generated tools come with features (e.g., GitHub Copilot) that help users to filter the inputs. Using these tools helps to reduce liability.

What companies must do

To address AI-generated code challenges, companies must establish clear-cut open source policies. Let's delve into a few essential steps.

- Establish open source policies and monitor the usage of open source components in the AI-generated code.
- Maintain business accounts, where administrators can set the configuration that helps to mandate the use of the AI tools.
- Implement a process for licence compliance, which includes internal audits of the applications and approval of the licences based on their obligations.
- Conduct internal audits before the release of the software to ensure that the code is free from critical OSS licences, that there are no major issues, and the applications are in line with OSS licence compliance.
- Educate and train development teams working on AI-generated tools about OSS licence compliance, licence obligations, attribution practices, and the implications of the different and critical OSS licences.
- Consult legal experts specialised in OSS licences and IP laws to ensure compliance with relevant guidelines and regulations. Seek their advice on the complex OSS licensing situations and any legal concepts in the context of AI-generated code or in the usage of AI generators.

Generative AI has revolutionised software development by automating the generation of code, thus accelerating developers' workflows. However, organisations must address OSS compliance when utilising these AI tools.

By understanding the OSS licences, implementing best practices, and leveraging automated tools, organisations can handle the code complexity and ensure OSS compliance requirements. All these approaches will promote a healthy OSS compliance ecosystem. END

References

- *https://plus.pli.edu/Details/Details?fq=id:(380453-ATL10)*
- *https://www.lawoftheledger.com/wp-content/uploads/sites/15/2023/04/AI-Code-Generators-Article_Part-1-0423.pdf*
- *https://www.lawoftheledger.com/wp-content/uploads/sites/15/2023/05/AI-Code-Generators-Article_Part-2-0523.pdf*
- *https://plus.pli.edu/Details/Details?fq=id:(380453-ATL10)#ID0EVD*

By: Lakshmi Teja

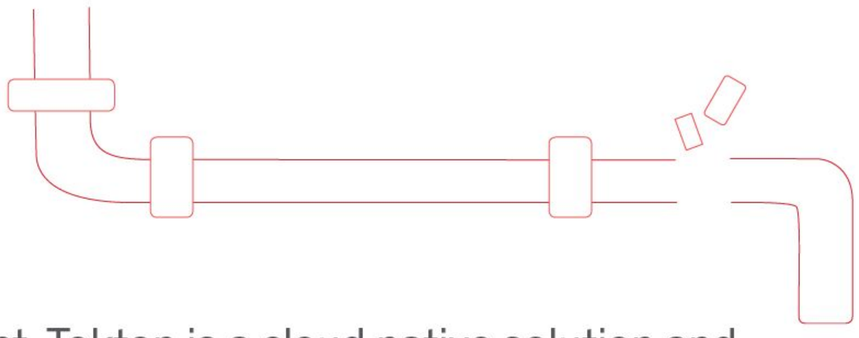
The author is an open source software licence consultant at Source Code Control Limited.

FOCUS

DevOps



Streamlining Software Delivery with Tekton Pipelines



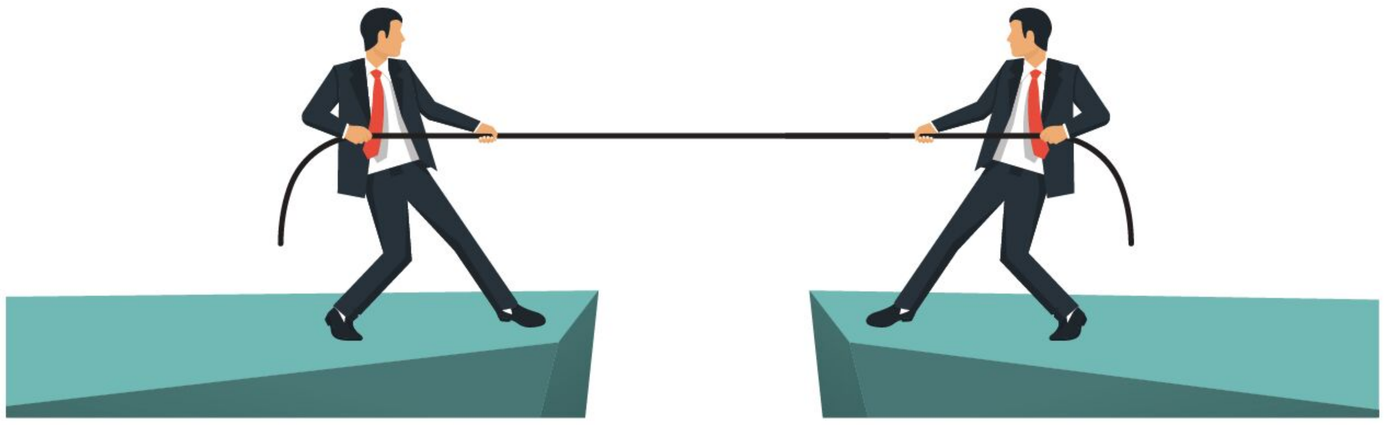
An open source project, Tekton is a cloud native solution and Kubernetes-native resource for building, testing, and deploying applications. It runs inside Kubernetes or its other variants, enhancing the efficiency of DevOps. *Read more on page.....21*

Enabling the Success of DevSecOps with AI



AI impacts DevSecOps in many positive ways — from accelerated testing and better security to quicker delivery of software. It promises to enhance every organisation's potential. *Read more on page.....33*

DevOps Vs Agile: What You Should Know



An open source project, Tekton is a cloud native solution and Kubernetes-native resource for building, testing, and deploying applications. It runs inside Kubernetes or its other variants, enhancing the efficiency of DevOps. *Read more on page.....49*

MLOps: Benefits, Applications and Popular Platforms

MLOps is a fusion of DevOps and machine learning. It brings the principles that govern DevOps to the field of machine learning, to help build models that improve the functioning of industries as varied as healthcare and e-commerce. There are a number of platforms that can be used for MLOps. We take a look at a few of these.
Read more on page.....29

Embracing Progressive Delivery in Kubernetes with GitOps

GitOps leverages Git's strengths to achieve reliable, automated, and consistent infrastructure and application management, making it well-suited for modern DevOps practices and containerised environments like Kubernetes.
Read more on page.....38

How DevOps Differs from Traditional IT, and Which One is Better

DevOps enhances collaboration, while traditional IT upholds stability. Knowing when to automate and when to rely on stability is key. Organisations must choose their strategy wisely for optimal outcomes in varied industries. A balance of both DevOps and traditional IT generally works well.
Read more on page.....52

Data Analysis with PandasAI, the Generative AI Python Library

With PandasAI, you can turn complex queries into simple conversations, unlocking insights that reshape your approach to data analysis. Let's see how...



Business users and data analysts can now become more productive at work than ever before, thanks to OpenAI's ChatGPT artificial intelligence tool. This new technology's capacity to facilitate the creation of SQL queries is a significant perk for business analysts. For years, business analysts have used SQL to query and analyse data. Learning and writing SQL is now easier than the conventional approach, thanks to the usage of AI technology, which can swiftly and automatically generate, edit, debug, and optimise SQL queries.

Python Pandas is an open source toolkit that gives data scientists and analysts the ability to manipulate and analyse data. In the preprocessing stage of machine learning and deep learning, the Pandas library is particularly popular. Now, with generative AI capabilities, the PandasAI module enables conversational data frame operation features for queries to the record set of a DataFrame.

Conversational query

PandasAI provides great strength to Pandas. A great amount of time is usually required to prepare a data set for final analysis. With the help of the PandasAI generative tool, it is possible to initiate the desired data analysis task on the raw data set with minimal preprocessing.

As PandasAI works with Pandas like a supplementary tool, one can apply generative tools on Panda's DataFrames and get back resultant DataFrames. OpenAI's PandasAI can engage in dialogue with a machine to get the desired results in DataFrame format without the need for writing lengthy queries or graphical Python codes.

Use of PandasAI: Installation and setup

The first task is to install PandasAI using the *pip install* command from the command line:

```
C:\mypython> pip install pandasai
```

Once installation is complete, the system is ready for action. Here, the Google Colab platform is used to demonstrate the role of PandasAI in data analysis. Start by calling *https://colab.research.google.com/* and then import the necessary modules as shown here:

```
import pandas as pd
import numpy as np
from pandasai import PandasAI
from pandasai.llm.openai import OpenAI
```

To upload the required input CSV file from the local directory, write down these file uploading instructions:

```
from google.colab import files
uploaded = files.upload()
```

Year	Dist_Name	RICE_AREA_1000_ha	RICE_PRODUCTION_1000_tons	RICE YIELD Kg_per_ha
1966	Burdwan	436.48	597.28	1368.4
1967	Burdwan	459.82	626.65	1364.61
1968	Burdwan	473.8	687.22	1454.51
1969	Burdwan	476.1	629.26	1322.7
1970	Burdwan	463.2	617.88	1332.78
1971	Burdwan	500.8	616.16	1233.81
1972	Burdwan	496	709.42	1430.28
1973	Burdwan	526.2	664.17	1262.2
1974	Burdwan	535.4	806.35	1506.07

Table 1: Rice production data of some districts of the state of West Bengal

	Year	Dist_Name	RICE_AREA_1000_ha	RICE_PRODUCTION_1000_tons	RICE_YIELD_Kg_per_ha
0	1966	Burdwan	436.48	597.28	1368.40
1	1967	Burdwan	458.82	625.65	1363.61
2	1968	Burdwan	472.80	687.22	1453.51
3	1969	Burdwan	476.10	629.26	1321.70
4	1970	Burdwan	463.20	616.88	1331.78

Figure 1: Rice production

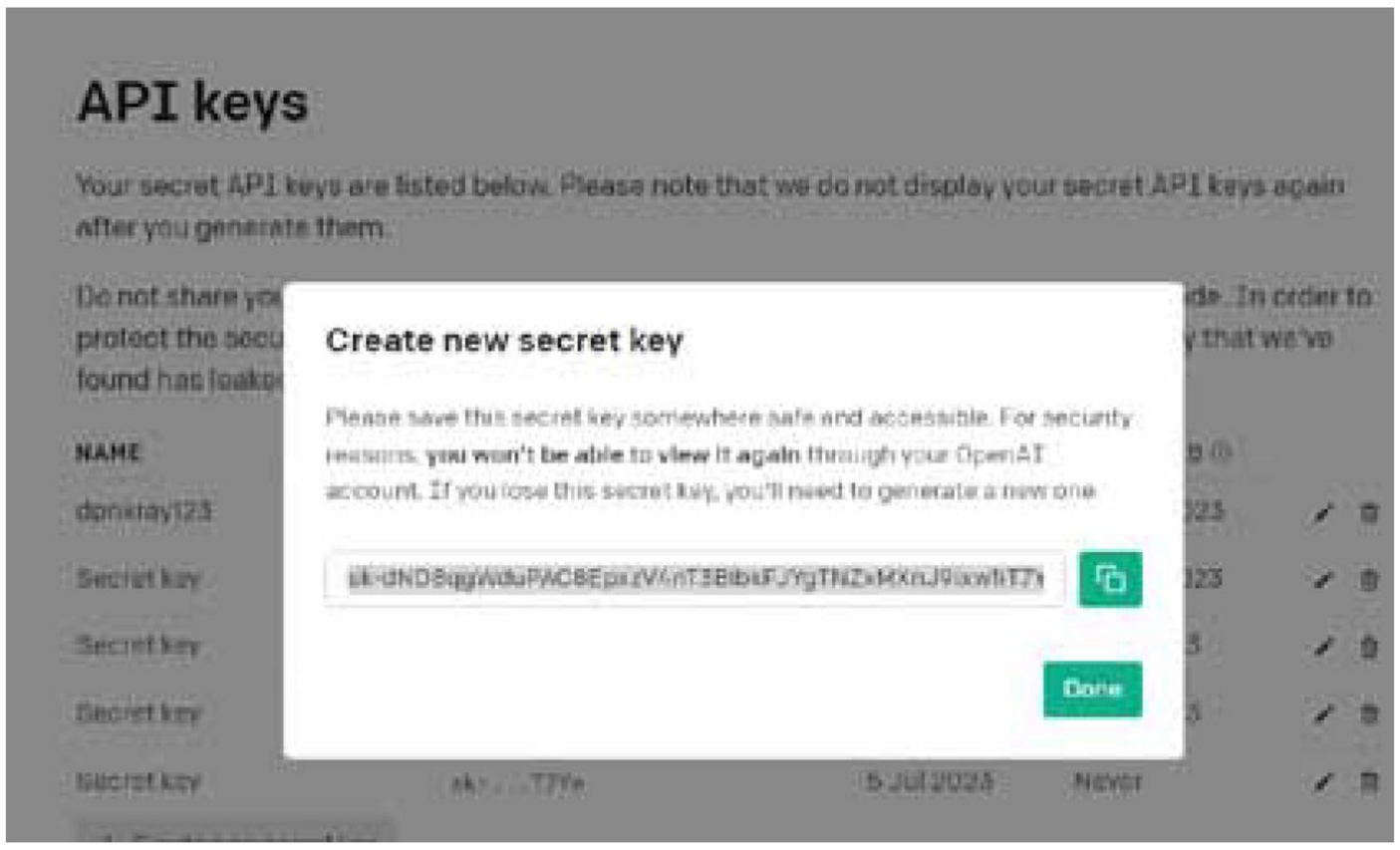


Figure 2: Generation of API key from ChatGPT

```
   Year Dist_Name  RICE_AREA_1000_ha  RICE_PRODUCTION_1000_tons  \
0  1966   Burdwan             436.48                     597.28
1  1967   Burdwan             458.82                     625.65
2  1968   Burdwan             472.80                     687.22
3  1969   Burdwan             476.10                     629.26
4  1970   Burdwan             463.20                     616.88

   RICE_YIELD_Kg_per_ha
0               1368.40
1               1363.61
2               1453.51
3               1321.70
4               1331.78
```

Figure 3: Output in tabular form

Once this command is executed, Python will request an input file. A CSV file named *CRISAT-WB_Rice.csv* has been uploaded in this instance. The file contains 364 record sets detailing rice production in various districts of West Bengal. Interested readers can download the file using the link provided at the end of the article.

Exploring data with PandasAI

After uploading the input CSV file, proceed to read the CSV file from memory:

```
import pandas as pd
import io
df = pd.read_csv(io.
StringIO(uploaded['ICRISAT-WB_Rice.csv'].
decode('utf-8')))

df.head()
```

The output is shown in Figure 1.

To call *pandasai()*, it is required to have a unique OpenAI API key. Obtain the key by creating an OpenAI account from *https://openai.com/blog/chatgpt*. From there it is possible to create an API key as shown here in Figure 2.

Once you have this key, you can call these two functions to start PandasAI:

```
oail = OpenAI(api_token="Insert your API-
KEY here")
pandas_ai = PandasAI(oali,
conversational=False)
```

For the 'True' value of the conversational option, PandasAI delivers a descriptive text record set, whereas it returns a tabular set of data sets for the 'False' value.

Working with Pandas DataFrame

The index of a record can be explored with the *pandas_ai()* call:

```
# Finding the index of a row using the value
of a column
result = pandas_ai(df, "What is the index
of 1982?")
print(result)
16
```

To display the first 5 records of the data set you can write this textual query:

```
response = pandas_ai(df, “Show 5 rows of data in tabular form”)
print(response)
```

The output is shown in Figure 3.

A more detailed conversational query can be written as follows:

```
response = pandas_ai(df, “Show 15 rows of data in tabular form   columns Year, DIST_NAME, RICE_PRODUCTION_1000_tons of DIST_NAME Hooghly RICE_PRODUCTION_1000_tons GREATER THAN 400”)
print(response)
```

The output is shown in Figure 4.

Similarly, we can also execute the following query over the DataFrame.

	Year	Dist_Name	RICE_PRODUCTION_1000_tons
164	1974	Hooghly	400.21
165	1975	Hooghly	404.47
167	1977	Hooghly	432.73
170	1980	Hooghly	510.82
173	1983	Hooghly	475.21
174	1984	Hooghly	472.88
175	1985	Hooghly	479.15
176	1986	Hooghly	458.60
177	1987	Hooghly	508.95
178	1988	Hooghly	584.78
179	1989	Hooghly	565.28
180	1990	Hooghly	610.69

Figure 4: Result of a detailed conversational query

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 364 entries, 0 to 363
Data columns (total 5 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Year                      364 non-null    int64
 1   Dist_Name                 364 non-null    object
 2   RICE_AREA_1000_ha         364 non-null    float64
 3   RICE_PRODUCTION_1000_tons  364 non-null    float64
 4   RICE_YIELD_Kg_per_ha      364 non-null    float64
dtypes: float64(3), int64(1), object(1)
memory usage: 14.3+ KB
```

Figure 5: Basic structures of a DataFrame

```
response = pandas_ai(df, “Show 15 rows of data in tabular form   columns Year, DIST_NAME and RICE_PRODUCTION_1000_tons DIST_NAME Bankura”)
print(response)
```

Data structure

It is possible to get all the basic structures of a DataFrame with simple textual instructions.

```
response = pandas_ai(df, “Show structure of data “)
print(response)
```

The shape of the DataFrame can also be explored with this instruction.

```
response = pandas_ai(df, “what is the shape of data?”)
print(response)

364 5
```

Data analysis

Overall basic statistics can easily be calculated with the following simple textual instruction:

```
response = pandas_ai(df, “what is the distribution  of RICE_PRODUCTION_1000_tans?”)
print(response)

count 364.000000
mean  641.068819
std   456.838868
min   51.490000
25%   289.392500
50%   493.310000
75%   848.247500
max   2145.290000
Name: RICE_PRODUCTION_1000_tons, dtype: float64
```

Skewness

To know the skewness of the distribution, you can use this simple query:

```
response = pandas_ai(df, “Show the skewness  of RICE_PRODUCTION_1000_tons”)
print(response)
1.2902333826377734
```

Graphics

Graphical representation is essential for effective data analysis. Distributions of different parameters

Stay **Connected.**
Stay **Informed.**
Stay **Ahead.**

EFY GROUP
YOURS SINCE 1969



SUBSCRIBE AND **SAVE**

TEAR OUT

ORDER FORM

TEAR OUT

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 ☐	2150 ☐	3000 ☐
		SAARC COUNTRIES (IN US$ BY AIR)	
	50 ☐	135 ☐	NA
		OTHER COUNTRIES (IN US$ BY AIR)	
	100 ☐	270 ☐	NA

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

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**

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.

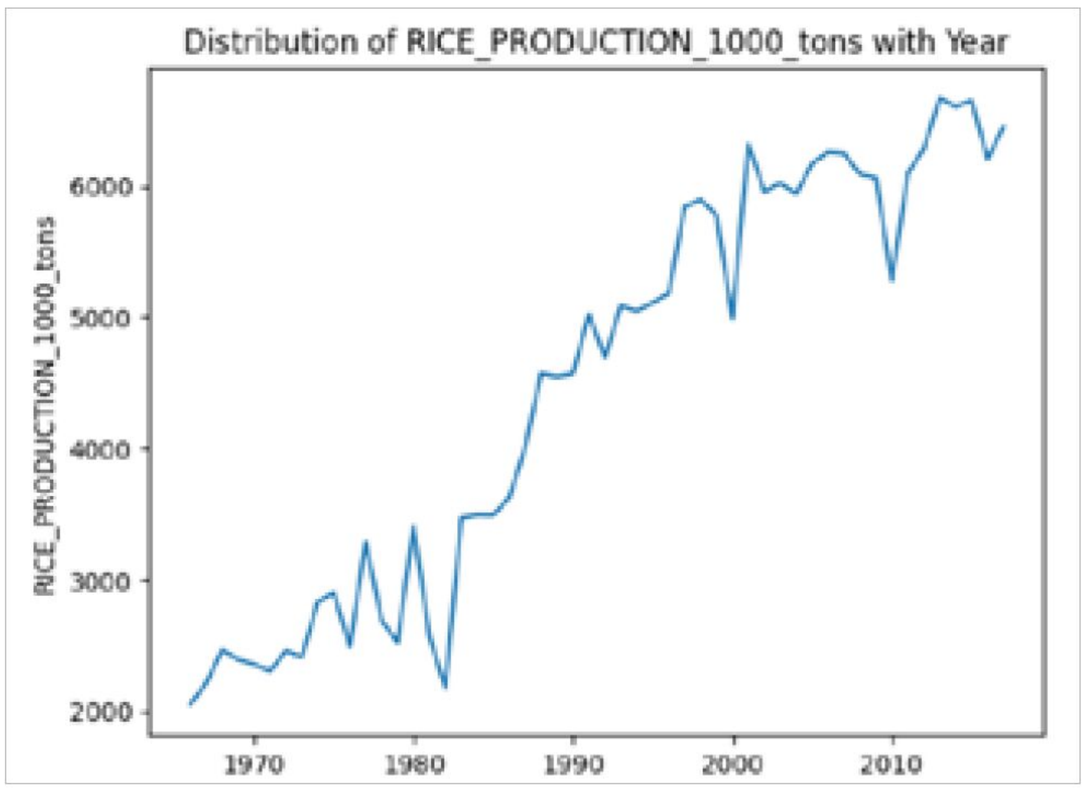


Figure 6: Distribution of rice production over the years

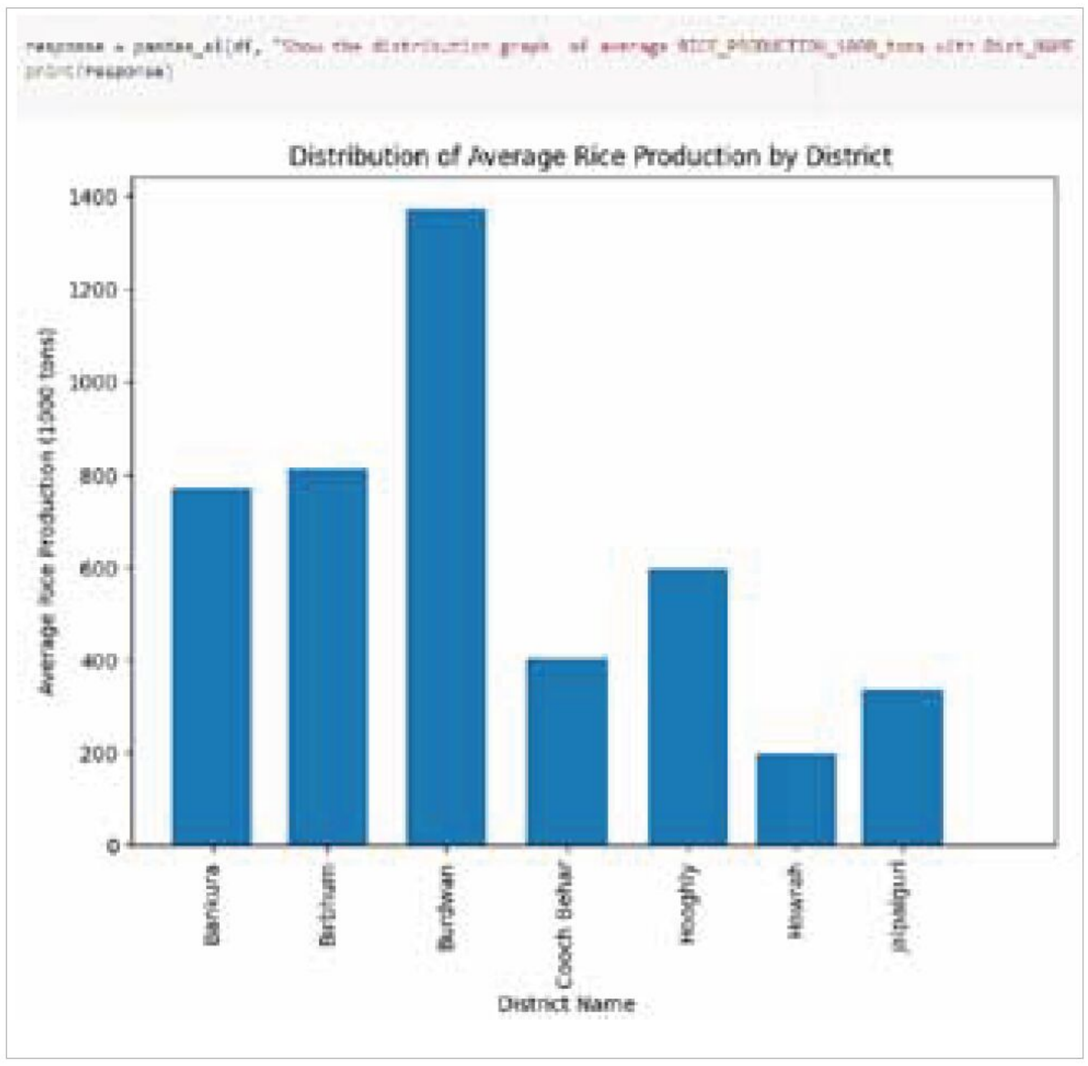


Figure 7: Distribution of district-wise rice production

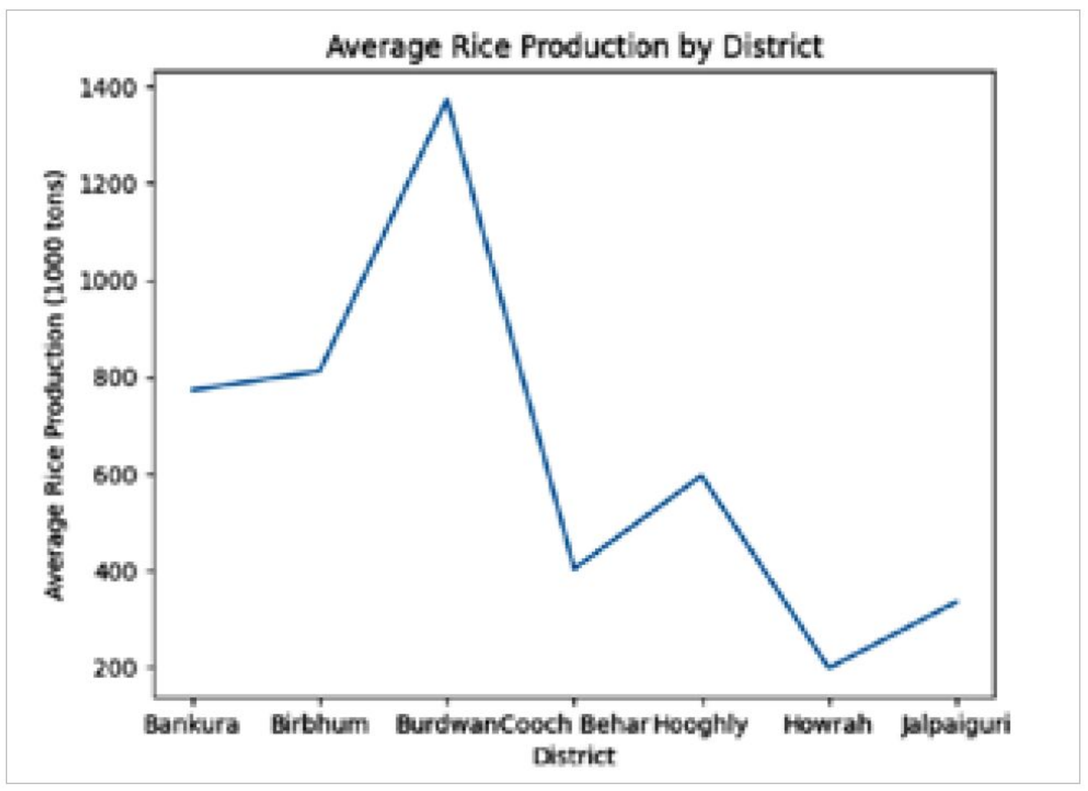


Figure 8: Line plot of average rice production by district

can easily be plotted with the *pandasai* module with the help of ChatGPT. For example, the distribution of rice production can be depicted as follows:

```
response = pandas_ai(df, “Show the distribution graph  of
RICE_PRODUCTION_1000_tons with Year”)
print(response)
```

Similarly, the distribution of rice production in a district can be plotted with an instruction like:

```
response = pandas_ai(df, “Show the distribution graph  of
RICE_PRODUCTION_1000_tons with Year DIST_NAME Hooghly”)
print(response)
```

A bar graph of the district-wise distribution of rice production can easily be drawn with an instruction like this:

```
response = pandas_ai(df, “Show the distribution graph  of
average RICE_PRODUCTION_1000_tons with Dist_NAME “)
print(response)
```

District-wise average rice production can be plotted like this**:**

```
df_sorted_rp=df.sort_values(by=’RICE_PRODUCTION_1000_
tons’)

response = pandas_ai(df_sorted_rp, “line plot average
RICE_PRODUCTION_1000_tons with DIST_NAME”)
print(response)
```

PandasAI, a supplementary tool based on ChatGPT, enhances data analysis in the Pandas environment. Its unstructured query language support makes data set analysis easier. Due to this textual informal approach, it is a paradigm change in data exploration and subsequent analysis. Here, we have focused on a few examples of the application of PandasAI on a data set. Readers can explore the several possibilities of this textual query-based data analysis. Programs and the data set are available at *https://drive.google.com/drive/folders/1gVvRcwDNDP4j_RbYdBm7Z0FKcUp95wNQ?usp=sharing*. Interested readers can download the data and explore it further. END

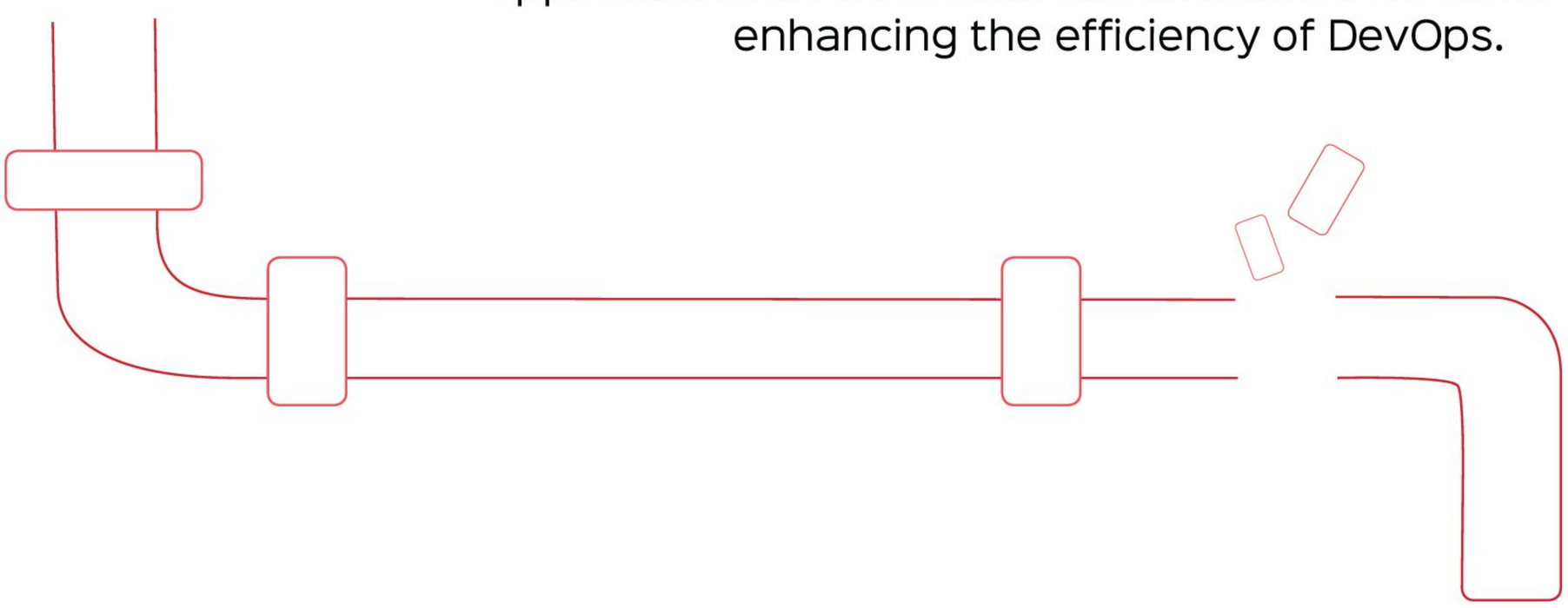
By: Dr Dipankar Ray

The author has more than 25 years of experience in software development. He has published two books on MATLAB and medical image processing (published by PHI Learning and SPD).



Streamlining Software Delivery with Tekton Pipelines

An open source project, Tekton is a cloud native solution and Kubernetes-native resource for building, testing, and deploying applications. It runs inside Kubernetes or its other variants, enhancing the efficiency of DevOps.



DevOps and Kubernetes (often referred to as DevOps with Kubernetes) complement each other to enable efficient and scalable software development, deployment, and management in modern cloud-native environments.

The combination of Kubernetes container orchestration capabilities and DevOps practices leads to improved collaboration, shorter development cycles, and enhanced application performance in modern cloud-native environments. By leveraging Kubernetes in a DevOps workflow, organisations can achieve faster, more reliable, and scalable software delivery.

Kubernetes offers several advantages when integrated with DevOps practices.

Agility and speed: Faster application deployments and updates enable swift responses to market demands and changes.

Scalability: Automatic horizontal scaling handles varying workloads efficiently.

Reliability and resilience: Self-healing capabilities and node management ensure continuous application availability.

Consistency and portability: Abstracting infrastructure allows consistent deployments across diverse environments.

Infrastructure as Code (IaC): Kubernetes resources defined as code enable version-controlled and repeatable infrastructure management.

CI/CD integration: It seamlessly integrates with CI/CD pipelines, streamlining the software delivery process.

GitOps: Version-controlled configurations ensure reliable and consistent application state management.

Microservices and service mesh: Suitable for microservices architectures, enhanced by service mesh for observability and security.

Resource utilisation: Optimises resource allocation, leading to cost savings and efficient cloud usage.

Observability: Extensive monitoring and logging capabilities provide insights into application and infrastructure health.

Multi-cloud and hybrid cloud support: Cloud-agnostic nature enables seamless deployments across various platforms.

Collaboration and communication: Encourages cross-functional collaboration, allowing developers to focus on application logic while operations handle infrastructure.

Problems of DevOps in Kubernetes

DevOps in Kubernetes comes with its own set of challenges due to the complexities of managing containerised environments and orchestrating microservices. Here are some common problems faced in DevOps with Kubernetes.

Steep learning curve: Kubernetes has a steep learning curve, especially for teams new to container orchestration and cloud-native technologies.

Infrastructure as Code (IaC): Writing and managing Kubernetes configurations as code (YAML manifests) can become complex, especially when dealing with large and distributed applications.

Versioning and rollbacks: Managing versioning and rollbacks of Kubernetes deployments can be challenging, especially when multiple microservices are involved.

Service discovery and load balancing: Properly managing service discovery and load balancing for microservices can be complex, especially in a dynamic and distributed environment like Kubernetes, OpenShift, ROSA, EKS, AKS and GKE.

Monitoring and observability: Setting up comprehensive monitoring and observability solutions for Kubernetes clusters and microservices can be daunting, requiring the integration of multiple tools.

Scalability and resource management: Scaling microservices dynamically while optimising resource allocation requires careful planning and continuous monitoring.

Secret management: Handling sensitive information like API keys, passwords, and certificates securely in Kubernetes can be tricky.

Application configuration: Managing application configurations across different environments consistently can be challenging, especially when multiple configurations are needed for various microservices.

Continuous delivery pipelines: Designing efficient and robust CI/CD pipelines for Kubernetes deployments requires a deep understanding of container images, Helm charts, and Kubernetes manifests.

Security and compliance: Ensuring security best practices and compliance standards are met in a containerised environment like Kubernetes can be complex.

Resource cleanup: Cleaning up unused resources and ensuring proper resource management in Kubernetes clusters can become a challenge, especially when dealing with dynamic scaling and frequent deployments.

DevOps tools

DevOps tools like Terraform, Ansible, Jenkins, AWS CodePipeline, AWS CodeDeploy, and Azure DevOps are widely used and powerful in traditional infrastructure and application management; they might face some challenges when used with Kubernetes. While these tools work well with infrastructure provisioning and configuring cloud based applications, they struggle when used with Kubernetes.

Paradigm mismatch: The DevOps tools mentioned before follow a procedural approach, whereas Kubernetes is declarative. This difference can lead to challenges in mapping workflows and resource management between these tools and Kubernetes.

Resource management: DevOps tools often have their own resource management mechanisms, which can conflict with Kubernetes resource management.

Kubernetes native concepts: Some of the previously mentioned tools might not fully leverage Kubernetes-native concepts and features, limiting their potential to take advantage of Kubernetes capabilities.

Kubernetes manifests: Kubernetes resources are defined in YAML manifests, which might not be natively supported by some DevOps tools, making it challenging to manage and update Kubernetes configurations.

Secret management: Handling Kubernetes secrets securely is crucial, and integrating this process seamlessly with the above DevOps tools might require additional effort.

Rolling updates and rollbacks: Kubernetes provides built-in support for rolling updates and rollbacks. Integrating these capabilities effectively with these DevOps tools can be complex, especially if the DevOps tools are not Kubernetes-aware.

Service mesh integration: Kubernetes service mesh technologies like Istio or Linkerd might not be well-integrated with these DevOps tools, leading to difficulties in managing advanced networking features.

Monitoring and observability: Some DevOps tools may not have native integrations with Kubernetes monitoring and observability solutions, making it harder to gain insights into the health and performance of Kubernetes deployments.

Dynamic environment changes: Kubernetes is designed to handle dynamic environments with ease, but this can pose challenges for some DevOps tools that are more suited to static environments.

Scalability and self-healing: While Kubernetes excels at dynamic scaling and self-healing, integrating these features effectively with some DevOps tools might require additional configurations and adjustments.

Continuous integration and continuous delivery (CI/CD) pipelines: These have become a cornerstone of modern software development practices. Among the numerous CI/CD solutions available, Tekton Pipelines stands out as a powerful and flexible platform, enabling teams to automate and streamline their software delivery workflows. We will explore the features, benefits, and applications of Tekton Pipelines, shedding light on how these have revolutionised the software development process.

Tekton

Tekton is a cloud native solution which is an open source project and Kubernetes-native resource for building, testing, and deploying applications. It consists of Tekton Pipelines, which provide the building blocks of supporting components, such as Tekton CLI and Tekton Catalog, that make Tekton a complete ecosystem. Tekton runs inside Kubernetes or its other variants like AWS ROSA.

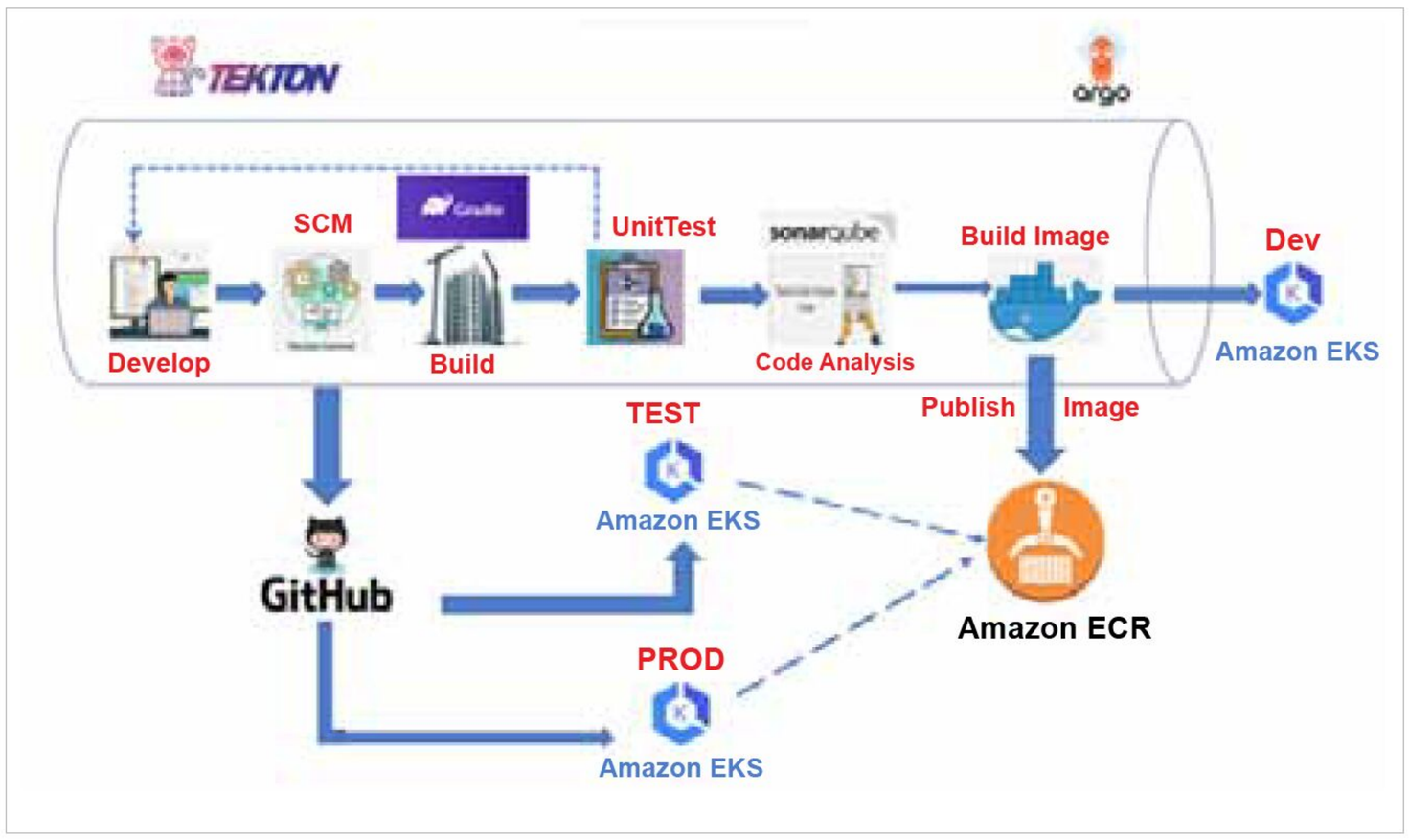


Figure 1: Tekton

Tekton provides several benefits to builders and users of CI/CD systems.

Customisable: Tekton entities are fully customisable, allowing for a high degree of flexibility. Platform engineers can define a highly detailed catalogue of building blocks for developers to use in a wide variety of scenarios.

Reusable: Tekton entities are fully portable; so once defined, anyone within the organisation can use a given pipeline and reuse its building blocks. This allows developers to quickly build complex pipelines without 'reinventing the wheel'.

Expandable: Tekton Catalog is a community-driven repository of Tekton building blocks. You can quickly create new and expand existing pipelines using pre-made components from the Tekton Catalog.

Standardised: Tekton installs and runs as an extension on your Kubernetes cluster and uses the well-established Kubernetes resource model. Tekton workloads execute inside Kubernetes containers.

Scalable: To increase your workload capacity, you can simply add nodes to your cluster. Tekton scales with your cluster without the need to redefine your resource allocations or any other modifications to your pipelines.

As a Kubernetes-native solution, Tekton seamlessly integrates with Kubernetes clusters. This allows developers to define and execute pipelines as Kubernetes Custom Resource Definitions (CRDs), making it easy to manage and monitor their pipelines alongside other Kubernetes resources.

Tekton pipelines embrace a modular approach to CI/CD, where tasks can be combined into customisable pipelines. This modularity fosters reusability and simplifies the maintenance of complex workflows. Teams can share and collaborate on tasks, promoting a culture of automation and efficiency.

These pipelines follow a declarative approach, allowing developers to specify what they want to achieve without being concerned with the underlying implementation details. This abstraction enhances pipeline maintainability and readability. Moreover, Tekton supports versioning, ensuring that pipelines remain consistent over time.

They leverage Kubernetes' scalability and isolation capabilities. Each task in a pipeline runs as a separate container, providing isolation and security. Teams can easily scale pipelines based on demand, ensuring optimal resource utilisation.

Tekton integrates well with a wide array of tools and platforms, including Kubernetes, Helm, Git, and more. This enables teams to create end-to-end delivery pipelines using their preferred tools while leveraging Tekton's core capabilities.

Tekton components

The Tekton ecosystem comprises various components that serve as the foundation for creating efficient and flexible CI/CD pipelines. These components seamlessly integrate within Kubernetes environments to orchestrate complex workflows with ease.

Tekton pipelines are the foundation of Tekton. They define a set of Kubernetes custom resources that act as building blocks from which you can assemble CI/CD pipelines.

Tekton triggers allow you to instantiate pipelines based on events. For example, we can trigger the instantiation and execution of a pipeline every time a PR is merged against a GitHub repository.

Tekton CLI provides a command-line interface called tkn, built on top of the Kubernetes CLI, which allows you to interact with Tekton.

Tekton dashboard is a web-based graphical interface for Tekton pipelines that displays information about the execution of your pipelines. It is currently a work-in-progress.

Tekton catalogue is a repository of high-quality, community-contributed Tekton building blocks — tasks, pipelines, and so on — that are ready for use in your own pipelines.

Tekton hub is a web-based graphical interface for accessing the Tekton Catalog.

Tekton operator is a Kubernetes operator pattern that allows you to install, update, and remove Tekton projects on your Kubernetes cluster.

Tekton chain provides tools to generate, store, and sign provenance for artifacts built with Tekton pipelines.

We can work with Tekton using tkn cli or Kubectl or Tekton API (available only for pipelines and triggers).

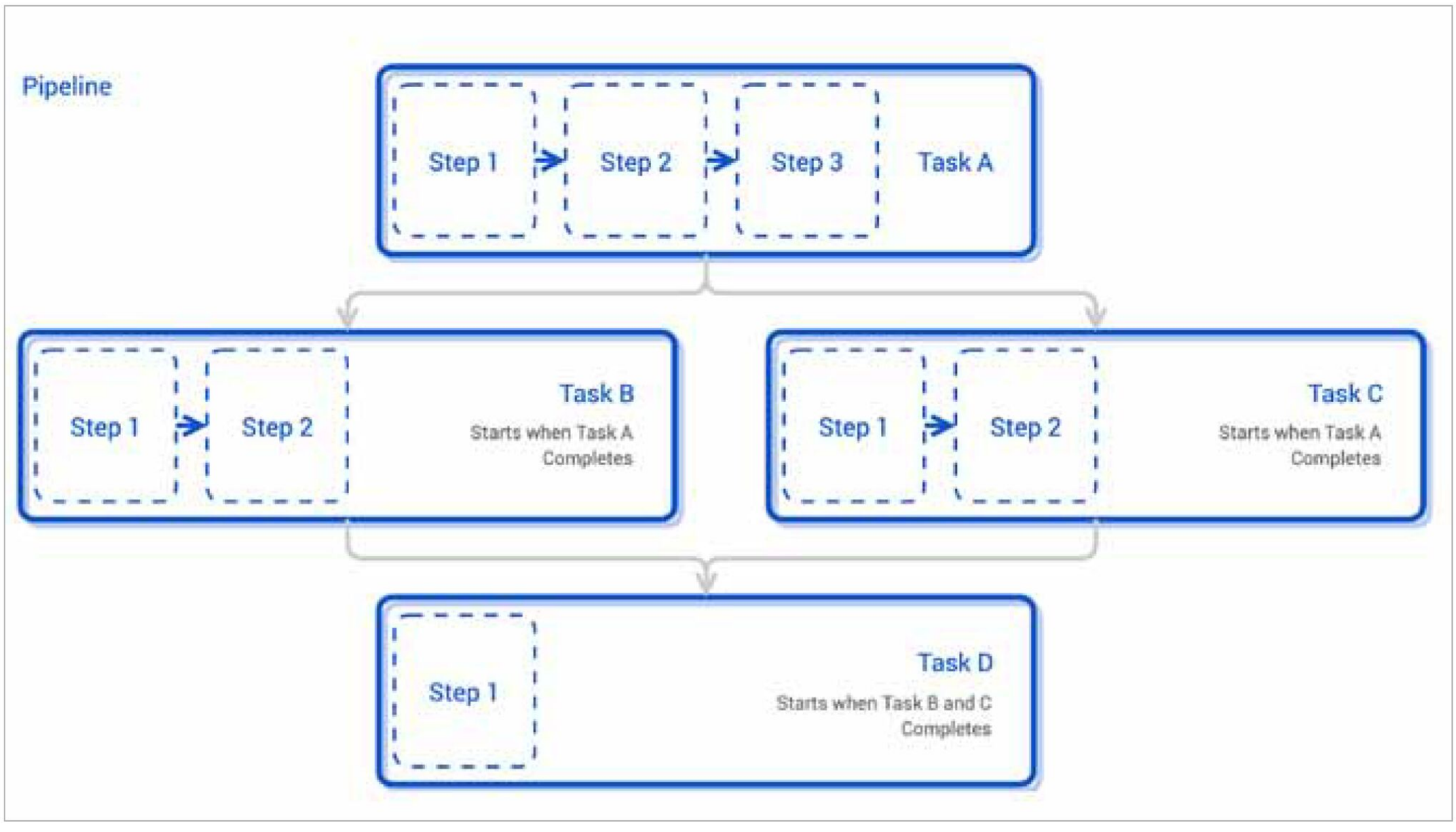


Figure 2: Pipeline

Anatomy of Tekton Pipelines

Here is a simplified breakdown of Tekton's anatomy.

Step is an operation in a CI/CD workflow, such as running some unit tests for a Python web app, or the compilation of a Java program. Tekton performs each step with a container image. Separate containers are created for each step.

At its core, Tekton is designed to simplify the CI/CD process by breaking down complex workflows into a series of smaller, reusable components known as tasks.

Task is a collection of steps in order. Tekton runs a task in the form of a Kubernetes pod, where each step becomes a running container in the pod. This design allows you to set up a shared environment for a number of related steps; for example, mounting a Kubernetes volume in a task will be accessible inside each step of the task. Each task will create one pod.

Taskrun is a specific execution of a task. Task needs either taskrun or pipelines to get executed. Taskruns can also be used to run a task outside a pipeline.

Pipeline is a collection of tasks in order. Tekton collects all the tasks, connects them in a directed acyclic graph (DAG), and executes the graph in sequence. In other words, it creates a number of Kubernetes pods and ensures that each pod completes running successfully as desired. Tekton grants developers full control of the process. You can set up a fan-in/fan-out scenario of task completion, ask Tekton to retry automatically should a flaky test exist, or specify a condition that a task must meet before proceeding.

Applications of Tekton Pipelines

Continuous integration: Tekton simplifies the process of building and testing code changes with automated CI pipelines. Developers can trigger these pipelines with each code commit, ensuring early detection of issues and promoting a consistent development workflow.

Continuous deployment: Tekton pipelines facilitate the deployment of applications to various environments, such as staging and production. Automated CD pipelines ensure that tested and approved code changes are safely and reliably pushed to production environments.

Canary and blue-green deployments: Using Tekton, teams can implement advanced deployment strategies like canary and blue-green deployments. These strategies allow for gradual rollouts and easy rollbacks, reducing the impact of faulty releases.

Custom workflows: Tekton's flexibility enables teams to create tailored workflows for specific use cases. For instance, they can orchestrate complex multi-step processes involving multiple services and dependencies.

Tekton pipeline creation

The easiest way of creating a Tekton pipeline is by using a Helm chart, but to

make it simpler we will go with simple YAML based declarations. We will write a separate article with a Helm chart sometime in future.

1. Create an EKS cluster in AWS cloud for the Tekton pipeline to run. Tekton pipelines will run inside a Kubernetes/OpenShift cluster; so this is a mandatory step.
2. Create a folder named IAC and create two sub-folders called AWS and GitOps.
3. The AWS folder will hold all infrastructure related workload creation tasks like S3 bucket, CloudWatch, EBS, etc.
4. The GitOps folder will consist of four sub-folders for different environments — *dev*, *test*, *stage* and *prod* folders. These folders will have the configuration definition as per their environment.
5. In the AWS folder, create cloudformation templates for creating security groups, load balancers, Apigateway, Route53 hosted zones, Alb controllers, etc.
6. Create a task for running the cloudformation template and name it as infratask.
7. Create a service account associated with roles and name spaces. Provide access tokens for Git, Docker Hub and EKS complete access.
8. Get access tokens for git access (read and write access), repository access and EKS deployment access with appropriate roles.
9. Create tasks for GITcheckout, build, package to Docker image and push it into image repository like Nexus or AWS ECR or Docker image repository. (You need to have tokens for this.)
10. Create a Tekton pipeline which consists of all the tasks sequentially or parallelly (as per your use case).
11. Create a pipeline run file which consists of the exact value of the git paths, storage volume, etc.
12. Docker containers are created using kaniko or buildah. These tasks are already available in Tekton Hub, which we can download and refer to in our pipeline.

Let us write a Tekton example code for pulling the Java code from the Git repository, build using Maven and package, create a Docker image, push it to Docker registry and deploy it into AWS EKS.

```
apiVersion: tekton.dev/v1beta1
kind: Pipeline
metadata:
  name: java-maven-docker-pipeline
spec:
  resources:
    - name: source-repo
      type: git
    - name: docker-image
      type: image
  tasks:
    - name: clone-repo
      taskRef:
        name: git-clone
      resources:
        inputs:
          - name: url
            resource: source-repo
          - name: revision
            resource: source-repo
      params:
        - name: path
          value: /workspace/source

    - name: build-with-maven
      taskRef:
        name: maven
      resources:
        inputs:
          - name: workspace
            resource: source-repo
      params:
        - name: PATH_TO_POM
          value: /workspace/source/path/
to/pom.xml
        - name: GOALS
          value: “clean package”

    - name: build-docker-image
      taskRef:
        name: buildah
      runAfter:
        - build-with-maven
      resources:
        inputs:
          - name: source
            resource: source-repo
      params:
        - name: DOCKERFILE
          value: /workspace/source/
Dockerfile
        - name: CONTEXT
          value: /workspace/source
        - name: IMAGE
          value: $(outputs.resources.
docker-image.url)

    - name: push-to-docker-registry
      taskRef:
        name: docker-push
      runAfter:
        - build-docker-image
      resources:
        inputs:
          - name: image
            resource: docker-image
      params:
        - name: TAG
          value: latest
        - name: IMAGE
          value: $(inputs.resources.
image.url)
        - name: REGISTRY
          value: DOCKER_REGISTRY
        - name: USERNAME
          value: your-docker-registry-
username
        - name: PASSWORD
          value: your-docker-registry-
password

    - name: deploy-to-eks
      taskRef:
        name: eks-deploy
      runAfter:
        - push-to-docker-registry
      resources:
        inputs:
          - name: source
            resource: source-repo
      params:
        - name: KUBECONFIG
          value: /workspace/.kube/config
        - name: CONTEXT
```

ACCELERATING DEVELOPMENT AND DEPLOYMENT OF OPEN SOURCE

20th edition









An EFY GROUP EVENT

tracks

Community
(India contributes)

AI & ML
(Opening up the Tech)

IT Infra
(Security, Storage & Cloud)

Developers' Meet

DevOps

Data & Bases:
Open Source Rocks!

Asia's #1 Conference on Open Source

20th Edition

OPEN SOURCE INDIA

19-20 October 2023

NIMHANS CONVENTION CENTER, BENGALURU

Register Now!

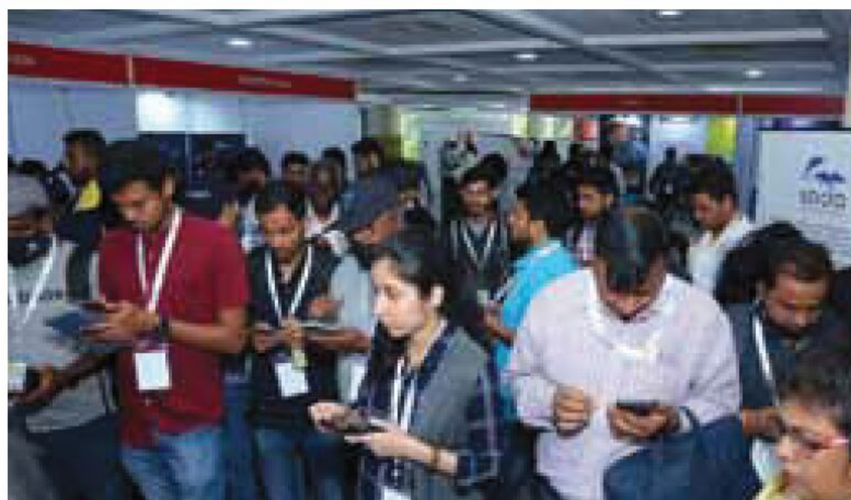
https://opensourceindia.in

www.OpenSourceIndia.in

For more details, call on +91 98111-55335 or write to info@opensourceindia.in







```
          value: eks-cluster
        - name: PATH_TO_K8S_YAML
          value: /workspace/source/
path/to/kubernetes-deployment.yaml
```

The necessary Tekton tasks (git-clone, maven, buildah, docker-push, and eks-deploy) are defined in your cluster (pulled from tektonhub), after you have set up the appropriate credentials for accessing the Git repository and Docker registry.

Additionally, the Kubernetes deployment YAML should be available in the Git repository and specified correctly in the PATH_TO_K8S_YAML parameter.

```
apiVersion: tekton.dev/v1beta1
kind: Pipeline
metadata:
  name: java-maven-docker-pipeline
apiVersion: Specifies the Tekton API
version being used.
kind: Indicates that this resource is a
Tekton Pipeline.
metadata.name: The name of the pipeline
resource.
spec:
  resources:
    - name: source-repo
      type: git
    - name: docker-image
      type: image
```

spec.resources: Defines the resources used by the pipeline.

resources.name: Names the resources. In this case, there's a Git resource named *source-repo* and an image resource named *docker-image*.

resources.type: Specifies the type of resource, which is either git for the Git resource or image for the Docker image resource.

These sections define the individual tasks that make up the pipeline:

```
tasks:
  - name: clone-repo
    taskRef:
      name: git-clone
resources:
  inputs:
    - name: url
      resource: source-repo
    - name: revision
      resource: source-repo
params:
  - name: path
    value: /workspace/source
```

tasks.name: Name of the task.

tasks.taskRef.name: Refers to the name of the Tekton task that will be executed (in this case, git-clone).

tasks.resources.inputs: Specifies the inputs required for this task. The URL and revision are inputs defined in the Git resource, and the path is set to the working directory for the task.

```
  - name: build-with-maven
    taskRef:
      name: maven
    resources:
      inputs:
        - name: workspace
          resource: source-repo
    params:
      - name: PATH_TO_POM
        value: /workspace/source/path/
to/pom.xml
      - name: GOALS
        value: "clean package"
```

This task (build-with-maven) runs the Maven Tekton task. It uses the Git resource *source-repo* as its workspace.

The Maven build is configured with the path to the *pom.xml* file and the goals 'clean package'.

The remaining tasks (build-docker-image, push-to-docker-registry, and deploy-to-eks) follow a similar structure. They reference specific Tekton tasks and define the resources and parameters required for each step.

```
Pipelinerun.yaml
apiVersion: tekton.dev/v1beta1
kind: PipelineRun
metadata:
  name: java-maven-docker-pipeline-run
spec:
  pipelineRef:
    name: java-maven-docker-pipeline
  resources:
    - name: source-repo
      resourceRef:
        name: your-git-resource-name
    - name: docker-image
      resourceRef:
        name: your-docker-image-
resource-name
```

To run the above pipeline:

Create pipeline: Apply the pipeline definition YAML to your cluster using *kubectl apply -f pipeline-definition.yaml*.

Create pipeline run: Apply the pipeline run YAML to start the execution of your pipeline using *kubectl apply -f pipeline-run.yaml*. Make sure you adjust the resource names (Git and Docker image resources) in the pipeline run YAML.

Monitor progress: Use commands like *kubectl get pipelineruns* and *kubectl describe pipelinerun <run-name>* to monitor the progress of your pipeline run. You can also use *kubectl logs -f taskrun <task-run-name>* to view the logs of individual tasks. END

Acknowledgements

I extend my gratitude to Balakrishnan Srinivasan (CTO - Hybrid Cloud Application Modernization Services, IBM India) for his encouragement to write this article, and Gopalakrishna U. (associate partner, IBM India) for providing the opportunity to gain real-time experience in this subject. Special thanks to Harsh Mehta (delivery manager, IBM India) for facilitating the release of this article for publishing.

By: Venkateswaran N.

The author is a lead architect, containerisation and modernisation, AWS practice, IBM India.

MLOps: Benefits, Applications and Popular Platforms



Image Source: https://www.freepik.com

MLOps is a fusion of DevOps and machine learning. It brings the principles that govern DevOps to the field of machine learning, to help build models that improve the functioning of industries as varied as healthcare and e-commerce. There are a number of platforms that can be used for MLOps. We take a look at a few of these.

Machine learning (ML) and DevOps are two critical fields that have been increasingly converging in recent years. DevOps refers to the specialised set of practices that combine software development (Dev) and IT operations (Ops) to shorten the systems development life cycle and provide continuous delivery with high software quality. When applied to ML, this convergence is often referred to as 'MLOps' (machine learning operations). MLOps aims to bring the principles of DevOps to the world of machine learning, facilitating the seamless integration of ML models into production environments while ensuring reliability, scalability, and reproducibility.

Key advantages of implementing MLOps

Using MLOps offers several advantages that significantly improve the development, deployment and management of machine learning models. Through streamlined processes and enhanced reproducibility, MLOps empowers organisations to harness their full potential.

Faster development cycles: MLOps automates and streamlines the end-to-end machine learning life cycle, including model training, testing and deployment. This results in faster development cycles, allowing data scientists and engineers to iterate and improve models more rapidly.

Improved model quality: MLOps promotes continuous integration and continuous deployment (CI/CD) practices for machine learning models. As a result, models can be tested thoroughly and deployed with greater reliability, leading to higher model quality and performance.

Reduced human errors: Automation in MLOps reduces the manual intervention required in the model deployment process, minimising the chances of human errors and ensuring consistency in deployments.

Enhanced collaboration: MLOps facilitates better collaboration between data scientists, software engineers, and operations teams. It creates a unified workflow where everyone can work together seamlessly, improving communication and knowledge sharing.



Figure 1: Key advantages with MLOps

Reproducibility and version control: MLOps enables version control and reproducibility of models and experiments. This ensures that models can be tracked, compared, and reproduced, making it easier to roll back to previous versions if needed.

Scalability and resource efficiency: MLOps provides tools and practices to optimise resource allocation, making model training and inference more scalable and efficient. It helps organisations make the most of their computational resources.

Model monitoring and maintenance: MLOps includes monitoring capabilities to track model performance in real-world environments. It can automatically detect concept drift or data drift and trigger retraining or updates as needed, ensuring models remain accurate and relevant.

Faster time-to-market: With automated processes and efficient workflows, MLOps reduces the time it takes to deploy machine learning models into production. This faster time-to-market can provide a competitive advantage for organisations.

Cost optimisation: MLOps practices help optimise the cost of running machine learning workloads. By automating resource allocation and monitoring, organisations can reduce unnecessary expenses.

Security and compliance: MLOps addresses security concerns in model deployment, ensuring that models and data are handled securely. It also helps organisations comply with industry regulations and data privacy requirements.

Flexibility in model deployment: MLOps allows organisations to deploy models across various platforms, including cloud, edge devices, and on-premises infrastructure, providing flexibility and adaptability to different deployment scenarios.

Continuous improvement: MLOps fosters a culture of continuous improvement. By integrating feedback from real-world usage, models can be continuously updated and enhanced, ensuring they remain effective and relevant over time.

MLOps offers numerous benefits that boost the efficiency, reliability, and effectiveness of machine learning deployments. It empowers organisations to deliver high-quality machine learning solutions more rapidly, enhancing their ability to leverage data-driven insights for business value and innovation.

Integration of MLOps with smart cities

MLOps offers tremendous potential in transforming the development and management of smart cities. By harnessing the power of data and AI-driven insights, MLOps enables efficient, scalable, and real-time solutions for the challenges faced in modern urban environments.

One of the key applications of MLOps in smart cities is in the field of traffic management. Machine learning algorithms can analyse vast amounts of traffic data, including live feeds from cameras, GPS data, and historical patterns. MLOps facilitates the continuous training and deployment of traffic prediction models, which can accurately anticipate congestion, optimise traffic flow, and even predict potential accidents. This results in smoother traffic management, reduced commute times, and improved overall safety for citizens.

Another significant use case is in waste management. By integrating MLOps practices, cities can optimise waste collection routes based on real-time data, including the fill-level of trash bins. This ensures that waste collection vehicles follow the most efficient paths, reducing fuel consumption and carbon emissions while keeping the city clean and hygienic.

MLOps can also be applied to enhance public safety and security in smart cities. Machine learning models can be trained to analyse video feeds

from surveillance cameras and identify suspicious activities or potential security threats. With MLOps, these models can be continuously updated and deployed to provide real-time monitoring, enabling law enforcement to respond swiftly to incidents and ensure the safety of residents.

MLOps can also help optimise energy consumption in smart cities. Machine learning algorithms can predict energy demand patterns, enabling cities to allocate resources efficiently and make data-driven decisions for energy distribution. MLOps facilitates the automatic updating of these predictive models, ensuring they remain accurate as energy usage patterns evolve.

It can be leveraged for environmental monitoring and sustainability initiatives. By analysing data from various sensors placed throughout the city, machine learning models can provide insights into air quality, water usage, and noise pollution levels. This data-driven approach empowers policymakers to implement targeted strategies for environmental conservation, and create greener and more sustainable urban spaces.

The use of MLOps in smart cities empowers urban planners, administrators, and citizens to make informed decisions based on real-time data and predictive analytics. It enhances the efficiency of city services, optimises resource allocation, and fosters a more livable, safe, and sustainable environment for the growing urban population. As smart cities continue to evolve, MLOps will undoubtedly play a central role in shaping the future of urban living.

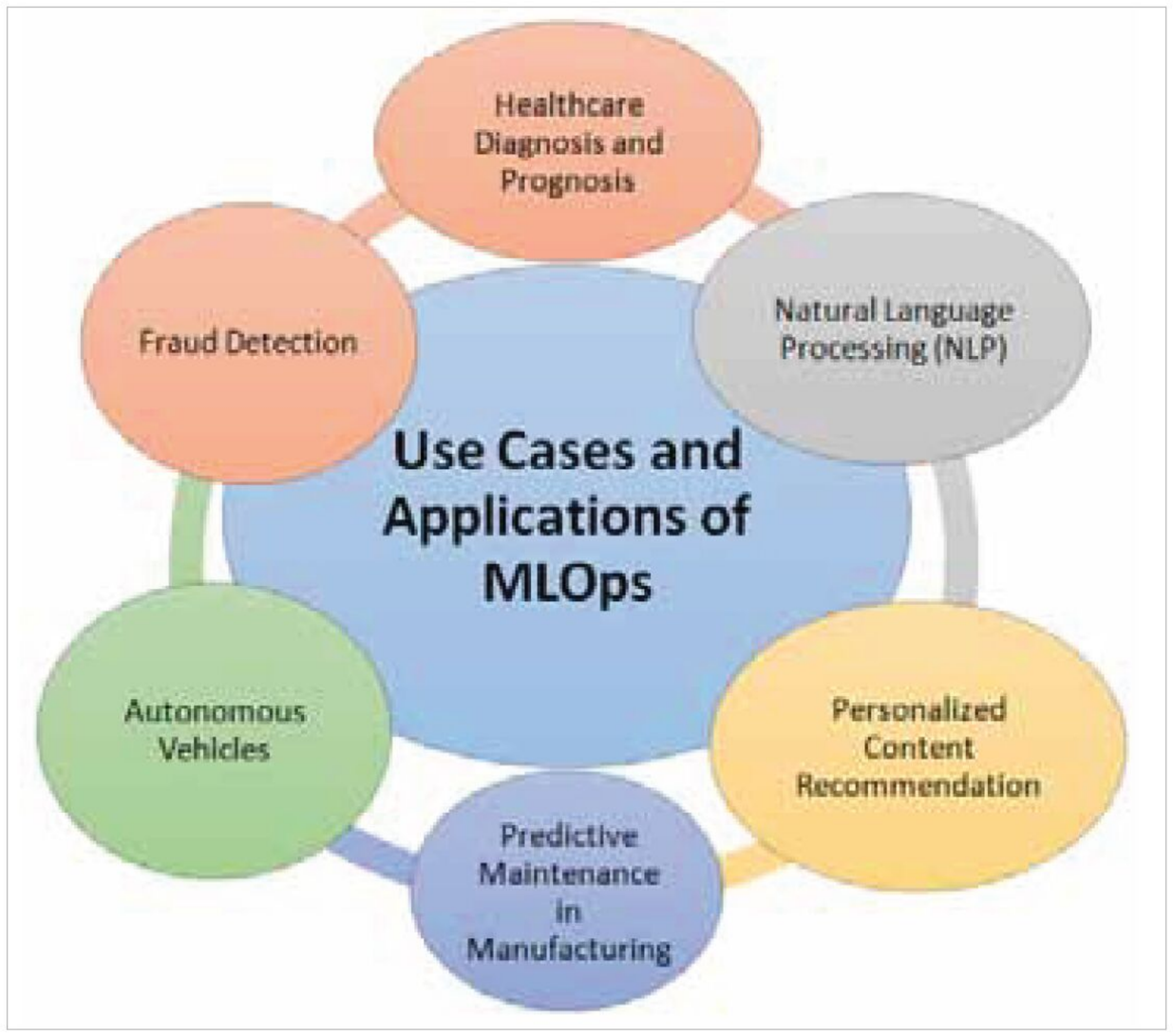


Figure 2: Use cases and applications of MLOps

Real world use cases and applications of MLOps

The practical application of MLOps across various industries demonstrates its transformative impact on diverse fields, from e-commerce and manufacturing to healthcare and finance. These real-world use cases exemplify how MLOps optimises processes, enhances decision-making, and ushers in a new era of efficiency and innovation.

Automated image classification in e-commerce: An e-commerce company utilises ML models to classify products automatically. MLOps helps in continuous training and deployment of these models as new product images are added or existing ones change. By automating the model deployment process, the company can maintain the accuracy of its classification system, ensuring customers receive relevant search results.

Predictive maintenance in manufacturing: A manufacturing plant employs ML models to predict machinery failures and schedule preventive maintenance. MLOps enables the seamless integration of these predictive models into their operational systems. This ensures that maintenance schedules are automatically updated based on real-time data, reducing downtime and increasing overall efficiency.

Fraud detection in finance: A financial institution uses ML models to detect fraudulent transactions. MLOps helps in automating the training and deployment of these models, ensuring that the fraud detection system remains up-to-date with the latest patterns and trends in fraudulent activities.

Healthcare diagnosis and prognosis: Hospitals implement ML models for medical image analysis, patient diagnosis, and treatment prognosis. MLOps ensures the continuous improvement and deployment of these models, making sure that medical professionals have access to the most accurate and reliable tools to aid in patient care.

Natural language processing (NLP) for customer support: Companies deploy NLP models to analyse and categorise customer support tickets. MLOps streamlines the process of updating and deploying these models as new customer queries and support ticket data are generated.

Autonomous vehicles: Companies working on autonomous vehicles require continuous development and deployment of ML models for perception, decision-making, and control systems. MLOps helps ensure that the most recent and optimised models are used, improving the safety and performance of autonomous vehicles.

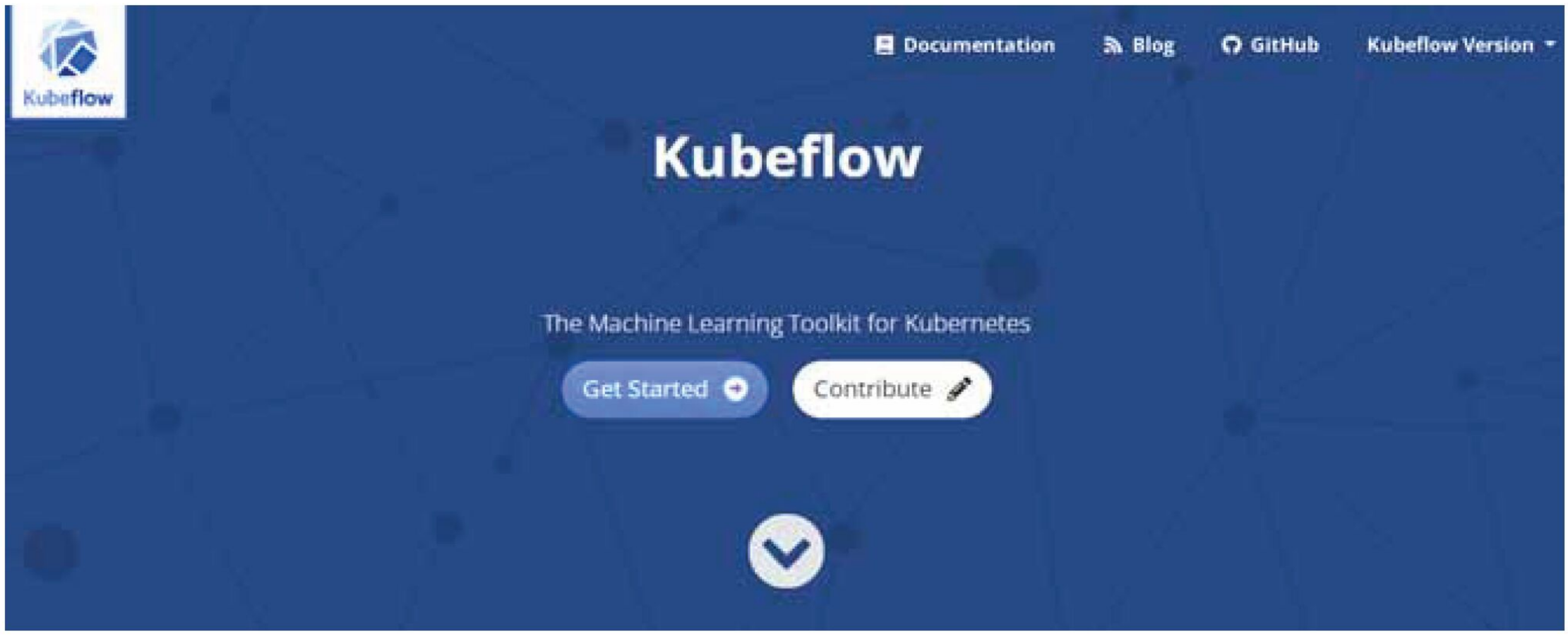


Figure 3: Kubeflow platform

Personalised content recommendation: Media and entertainment platforms utilise ML models to recommend personalised content to users. MLOps enables the seamless delivery of content recommendations in real-time, enhancing user engagement and satisfaction.

In all these cases, MLOps plays a crucial role in automating the ML model life cycle, including model training, testing, deployment, monitoring, and updates. This streamlines the development process, reduces manual errors, enhances collaboration between teams, and allows organisations to deliver high-quality ML-powered applications more efficiently and reliably.

Prominent platforms for MLOps

Many platforms offer a robust ecosystem to streamline machine learning workflows. Two major ones are Kubeflow and TensorFlow Extended.

Kubeflow

URL: https://www.kubeflow.org/

This is an open source platform built on Kubernetes, designed to streamline the deployment, scaling and management of machine learning workloads. It provides tools for building, training, and deploying models in a consistent and reproducible manner. Kubeflow leverages Kubernetes' container orchestration capabilities to enable seamless scalability of ML workloads. This means that as the demand for machine learning models grows, Kubeflow can efficiently allocate resources and scale the infrastructure to meet it. This elasticity ensures that ML models can handle various workloads, from small-scale experimentation to large-scale production deployments.

In MLOps, it is vital to have a well-documented and version controlled process for model training. Kubeflow provides tools and components that allow data scientists to package their ML code, dependencies, and configurations in a consistent manner using containers. This ensures that models can be easily reproduced and versioned, facilitating collaboration among teams and reducing the risk of model discrepancies in production.

TensorFlow Extended (TFX)

URL: https://www.tensorflow.org/tfx

Developed by Google, TFX is an end-to-end platform for deploying production machine learning pipelines. It includes components for data validation, preprocessing, model training, and serving. It plays a crucial role in the implementation of MLOps by enabling the end-to-end management of machine learning workflows. By seamlessly integrating TensorFlow, TFX facilitates the efficient development, deployment, and monitoring of machine learning models in real-world applications. At its core, TFX is designed to address the complexities associated with scaling and operationalising machine learning projects. It encompasses a collection of modular components, each serving a specific function within the pipeline. These components work cohesively to automate the machine learning life cycle, from data ingestion to model serving and beyond.

Other platforms for MLOps

There are other, lesser-known platforms too that play a vital role in shaping the MLOps landscape.

MLflow

URL: https://mlflow.org/

This open source platform facilitates the management of the complete machine learning life cycle. MLflow allows you to track experiments, package code into reproducible runs, and share and deploy models.

Continuted to page...35

Enabling the Success of DevSecOps with AI

AI impacts DevSecOps in many positive ways — from accelerated testing and better security to quicker delivery of software. It promises to enhance every organisation's potential.



In any organisation with a technology-driven IT department, the pursuit of efficiency, agility and cost-effective high quality software development is an ongoing quest. This desire for rapid software development, aligned with the ever-increasing urgency of business stakeholders, has led to the evolution of modern DevSecOps practices. This article delves into how AI is reshaping the landscape of DevSecOps.

Elevating development and testing with DevSecOps

Successful digital transformation stories have a common denominator — the adoption of modern DevSecOps practices. This transformative process encompasses build-test-release operations, prioritising speed, quality and value creation.

Development and testing teams now face the challenge of delivering accurate, high quality products that adhere to composable architecture and cloud-based application requirements, along with rapid validation and deployment of the solution. The foundation of this efficiency lies in continuous integration (CI) and continuous delivery (CD). CI standardises development practices, while CD ensures rapid deployment with necessary libraries, configurations and provisioning. These practices, collectively known as DevOps, have proven effective for organisations embracing them.

The testing of developer code at unit level could be integrated within the CI services of most readily available products, open source or otherwise. However, when it came to functional, system and other types of testing, organisations had built manual or siloed automations. These required significant rework with every change in code along with manual interventions in execution, validation and the reporting aspects of testing. The very nature of separation of duties and the function of testing required a different approach to bring it all together.

Table 1 lists the different types of testing and their roles in any given software development initiative. These testing types are not comprehensive; however, the intent is to showcase the various types of testing involved in a typical application development life cycle and the responsibilities

spread across various experts within the organisation. After all, standardising, managing and getting software out of the door is a complex exercise.

Modern enterprises expect near-real time (if not real-time) access to applications as they are developed. Gone are the days waiting for a URL update for six months because it's the norm. Integrating security within CI and CD processes has resulted in the evolution of DevSecOps. Figure 1 shows the integration of typical software development, testing and deployment, and the operational journey of code. If planned along with the application design and architecture, DevSecOps can be automated more quickly as all value propositions can focus only on gaining efficiency and forego all repetitive mundane tasks of building, testing and deploying.

The security (Sec) part of DevSecOps automates all types of testing as part of code build and not just security. The code being validated against testing first goes through unit, functional and integration testing to ensure it does what is expected of it, and then the security is validated before the code is deployed.

Now that we have established what DevSecOps is and the automation of code development, testing, deployment and monitoring aspects, let's look at some of the key benefits of adopting AI in DevSecOps:

- Improved security and reporting
- Improved reliability and resiliency of automation
- Ease of access to contextual data across the development life cycle
- Enhanced resource management with ability to respond automatically
- Ambiguity and anomaly detection with reliable pattern analysis
- Improved collaboration with AI connecting the knowledge bases

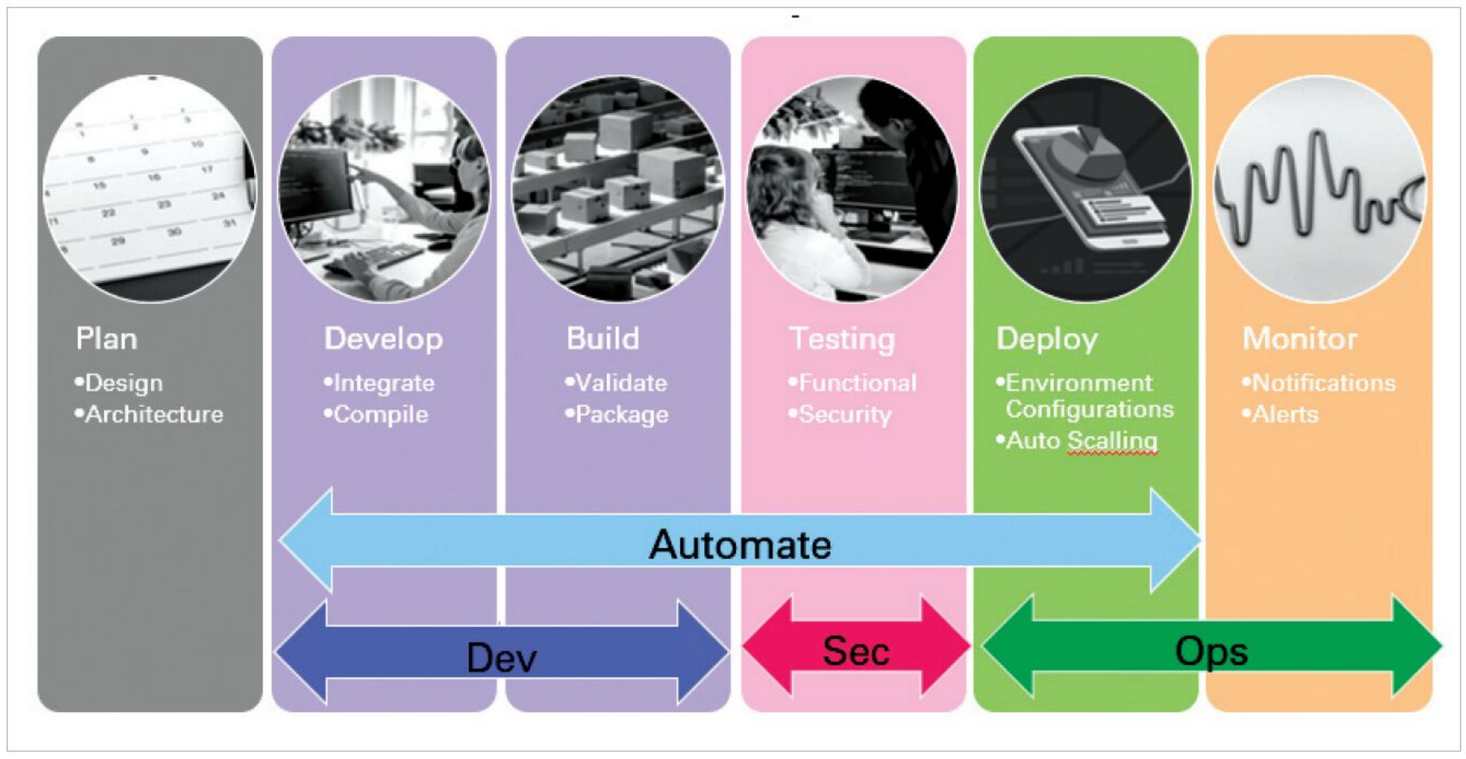


Figure 1: Automated DevSecOps orchestration

Testing type	Description	Responsibility of
Unit testing	Test individual modules of application in isolation to validate code is executing as intended	Developer
Functional testing	Test a specific functionality in the system to validate the code is executing and responding to user actions	Tester
System acceptance testing (SAT)	Also known as integration testing that involves overall testing of systems of many sub-systems/ applications or elements.	Domain experts and business users
User acceptance testing (UAT)	Validate application life cycle before go-live	Business users
Security testing	Process intended to reveal flaws in the security mechanisms	Security tester

Table 1: Types of testing and who is responsible for them in an organisation

The power of AI-driven chatbots

Chatbots have become a common tool for everyday users of technology across all types of devices. The younger generation thrives in the world of social media apps that are designed to respond to the briefest of interests with specific information. By integrating knowledge repositories, information security policies and regulations, test databases and generative AI capabilities, a powerful chatbot can offer valuable insights. For example, a developer can request the bot to compile specific compliance mandates so the person can understand how to code according to them. Or a tester can request what types of tests have been performed historically against a specific code module and what the outcome of those test executions are, to help validate the stability of the code. This can also be used by project managers to learn the effort and time it has taken for specific modules to be updated in the past, and forecast more accurately the release date of the application.

When only a select few individuals in an organisation know how to respond to specific scenarios or incidents, it is typically known as tribal knowledge. This knowledge can be automated by creating a feedback loop across the DevSecOps pipeline, which is data-driven. Here, events are funnelled to a previous process with AI to help learn or adapt to new information.

Many organisations are rapidly adopting AI through their machine learning platforms built on TensorFlow, PyTorch and others. These platforms enhance software development, testing and the delivery teams' speed and agility.

Change is never easy and is often met with scepticism. Yet, high-performing organisations adopt and redefine how they use technology and manual interventions by automating every possible process. While AI promises to minimise developer functions, there are some worries about loss of jobs. However, by embracing AI with proper planning and establishing governance policies, organisations can enhance decision-making, efficiency and productivity, without necessarily eliminating human roles. END

References

- DevOps.com | AI; *https://devops.com/category/blogs/ai/*
- OpenAI platform; *https://platform.openai.com/overview*
- Investing in AI: How MLOps Drives Enterprise AI Wins; *https://devops.com/investing-in-artificial-intelligence-how-mlops-drives-enterprise-ai-wins/*
- 13 open source projects transforming AI and machine learning; *https://www.infoworld.com/article/3673976/13-open-source-projects-transforming-ai-and-machine-learning.html*
- An end to end machine learning platform; *https://www.tensorflow.org/*
- PyTorch 2.0 next-gen platform for eager-mode dev and user experience; *https://pytorch.org/get-started/pytorch-2.0/*

By: Bala Kalavala

The author is a technical architect, evangelist, thought leader, and sought-after keynote speaker. He currently works as digital transformation business development executive in a global technology consulting firm.

Disclaimer: This article expresses the views of the author and not of the organisation he works in.

Continued from page...32

DVC (Data Version Control)

URL: https://dvc.org/

This version control system is designed for managing machine learning models, data sets, and experiments efficiently.

Apache Airflow

URL: https://airflow.apache.org/

While primarily used for data orchestration and workflow automation, Airflow can also be leveraged for MLOps to schedule and monitor model training and deployment tasks.

Metaflow

URL: https://metaflow.org/

Developed by Netflix, Metaflow is a data science framework that assists in building and managing real-life data science projects. It allows easy integration of code, data, and models.

Knative

URL: https://knative.dev/

Not specific to MLOps, Knative is an open source platform that extends Kubernetes to support serverless workloads, making it useful for managing serverless machine learning deployments.

CML (Continuous Machine Learning)

URL: https://cml.dev/

An open source library that integrates with Git and GitHub Actions to enable continuous integration and delivery for machine learning projects.

Seldon Core

URL: https://www.seldon.io/tech/products/core/

Built on Kubernetes, Seldon Core helps deploy and manage machine learning models at scale, allowing you to create production-ready inference graphs.

Apache Spark

URL: https://spark.apache.org/

While primarily a Big Data processing engine, Spark's MLlib library provides various tools for building and managing machine learning pipelines.

Research in the domain of MLOps can delve into identifying and analysing the best practices, methodologies and frameworks for integrating machine learning and DevOps effectively. This could involve studying existing MLOps tools, platforms, and practices to understand their strengths, weaknesses, and areas of improvement. Research may also explore techniques for automating model training, testing, and deployment pipelines.

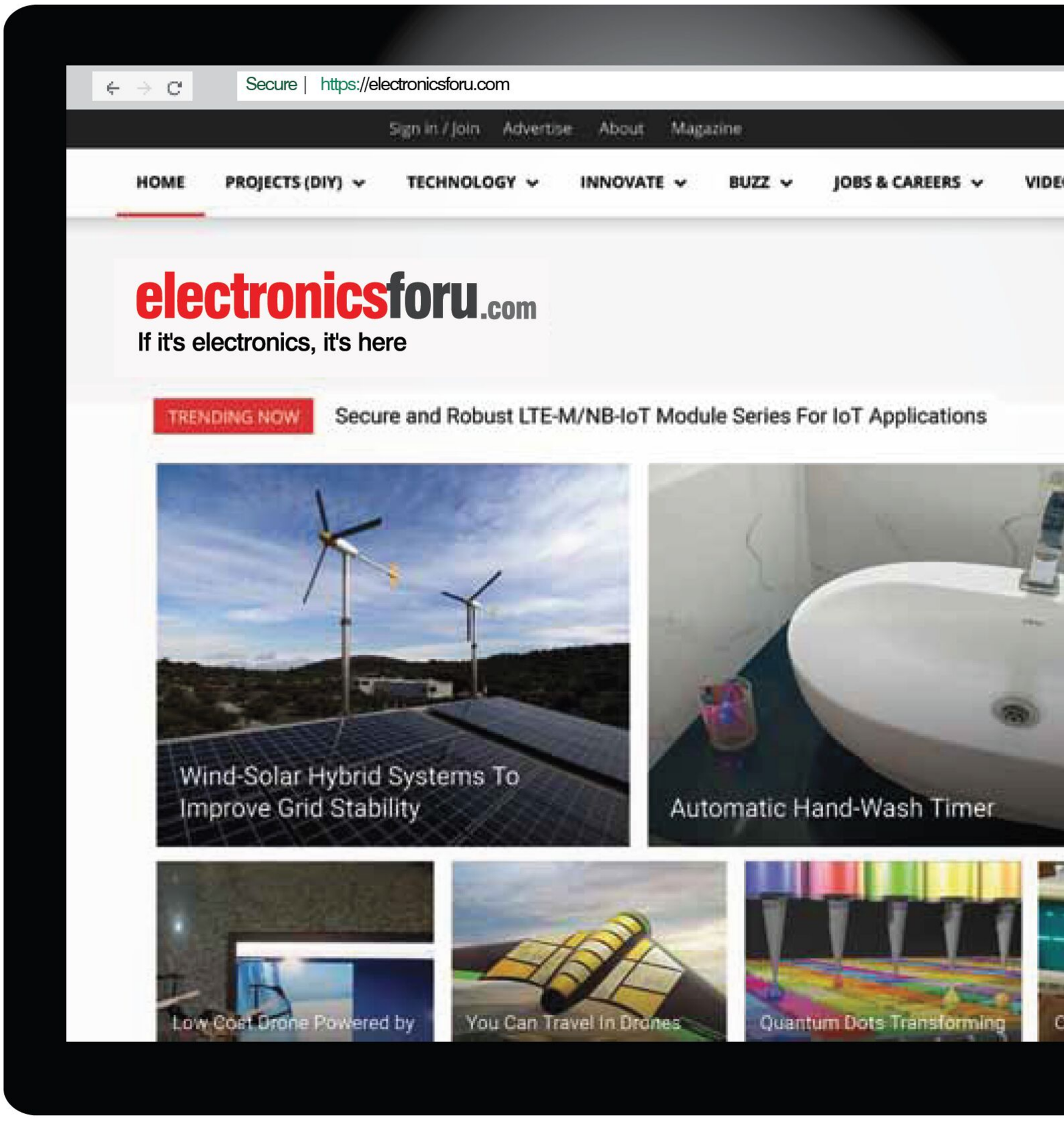
Research could also address the security and governance challenges associated with deploying ML models in production environments. This may involve exploring methods to secure models, data, and infrastructure, as well as adhering to regulatory compliance.

Researchers can contribute to enhancing the efficiency, reliability, and ethical implications of deploying ML models in real world settings, making them more accessible and beneficial to society. 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.

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

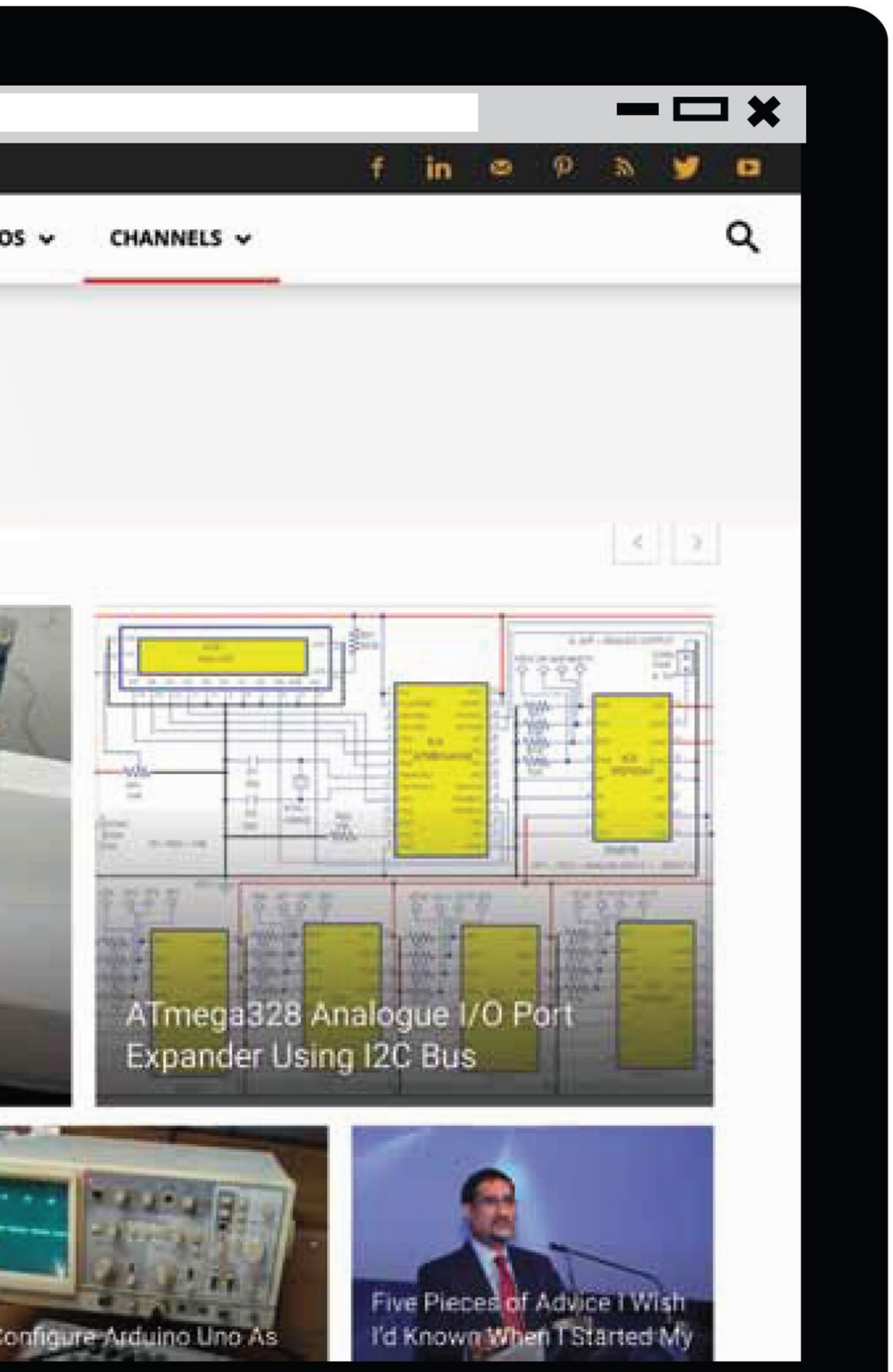
READERS

- 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

Embracing Progressive Delivery in Kubernetes with GitOps

GitOps leverages Git's strengths to achieve reliable, automated, and consistent infrastructure and application management, making it well-suited for modern DevOps practices and containerised environments like Kubernetes.



Kubernetes is an open source container orchestration platform that automates the deployment, scaling, and management of containerised applications. It provides a powerful and flexible solution for managing containerised workloads in a highly efficient and scalable manner.

Kubernetes plays a central role in modern cloud-native application development and has become the *de facto* standard for container orchestration due to its powerful features, scalability, and wide adoption within the industry.

Key features of Kubernetes

Here are the key features that distinguish the versatility of this platform.

Container orchestration: Kubernetes manages the deployment and operation of containers, allowing developers to focus on application logic rather than infrastructure details.

Automatic scaling: Kubernetes can automatically scale the number of container replicas based on defined resource utilisation metrics, ensuring applications are highly available and can handle varying loads.

Service discovery and load balancing: Kubernetes provides built-in service discovery and load balancing mechanisms, making it easy for containers to communicate with each other and distribute incoming traffic.

Self-healing: Kubernetes continuously monitors the health of containers and can automatically restart or replace unhealthy containers to maintain application availability.

Rollouts and rollbacks: Kubernetes supports rolling updates, enabling seamless application updates without downtime. It also allows easy rollbacks to a previous version in case of issues.

Storage orchestration: Kubernetes manages persistent storage for containers, enabling applications to store and access data that persists across container restarts.

Configurations and secrets management: Kubernetes allows you to define configuration data and secrets separately from the application code, promoting security and easy management.

Horizontal and vertical scaling: Kubernetes supports both horizontal scaling (increasing the number of replicas) and vertical scaling (increasing container resources like CPU and memory) to meet varying workload demands.

Batch processing and Cron jobs: Kubernetes provides support for batch processing workloads and scheduling periodic jobs using Cron expressions.

Multi-cloud and on-premises support: Kubernetes is cloud-agnostic and can run on various cloud providers as well as on-premises data centres.

Architecture of Kubernetes

Understanding the architectural foundation of the Kubernetes platform is essential for harnessing its full potential in orchestrating modern applications.

Master node: The control plane consists of several components running on the master node, including the API server, etcd (key-value store), controller manager, and scheduler. The master node manages the overall state of the cluster and makes decisions about scheduling, scaling, and maintaining desired state.

Worker nodes: Worker nodes are the worker machines responsible for running containers. Each worker node runs a container runtime (e.g., Docker), and the Kubernetes agent called kubelet, which communicates with the master node.

Pods: The smallest deployable units in Kubernetes are pods. A pod can contain one or more tightly coupled containers sharing the same network name space and volumes.

Services: Services provide stable endpoints to access pods and enable load balancing across replicas of an application.

Deployments and ReplicaSets: Deployments and ReplicaSets are used to manage the life cycle of pods, ensuring a specified number of replicas are running and enabling updates and rollbacks.

Deploying applications in Kubernetes

Each application deployment has specific requirements, such as configuring environment variables, setting resource limits, and handling dependencies. Kubernetes offers a flexible and robust platform to manage a variety of containerised applications, making it suitable for various use cases, from microservices to large-scale applications.

Deploying applications in Kubernetes involves several steps to ensure that your containerised applications are up and running in the cluster. Here's a high-level overview of the process.

***Containerise your application*:** The first step is to package your application into a container image. This typically involves writing a Dockerfile that specifies the application's runtime environment and dependencies.

Build and push the container image: Once the Dockerfile is ready, you build the container image using a container runtime like Docker. After building, you push the image to a container registry like Docker Hub, Google Container Registry, or Amazon Elastic Container Registry (ECR).

Create Kubernetes manifests: Kubernetes uses YAML files called manifests to define the desired state of your application. Manifests include details such as the container image, resource requirements, environment variables, networking, and more.

Deployments or StatefulSets: To manage your application's replicas, you can use either deployments (for stateless applications) or StatefulSets (for stateful applications that require stable network identities).

Services: To expose your application to the network and enable load balancing, you create a Kubernetes service. The service provides a stable IP address and DNS name for your application.

Persistent storage (if needed): If your application requires persistent storage, you can define PersistentVolumeClaims (PVCs) and PersistentVolumes (PVs) to provide durable storage to your application.

Namespaces: Namespaces allow you to logically divide your cluster, isolating different applications and teams from each other.

Apply manifests to the cluster: Once your manifests are ready, you apply them to the Kubernetes cluster using the 'kubectl apply' command or through a CI/CD pipeline.

Monitoring and observability: Set up monitoring and logging solutions to track the health and performance of your applications in the cluster. Popular tools include Prometheus, Grafana, and ELK stack.

Continuous deployment (optional): Implement a CI/CD pipeline to automate the deployment process, allowing changes to be automatically rolled out to the cluster when code is pushed or merged.

Testing and validation: Before deploying to production, thoroughly test your application in staging or testing environments to catch any potential issues.

Scaling and updating: Kubernetes provides mechanisms to scale your application based on demand, as well as rolling updates and rollbacks for seamless application updates.

Pain points in Kubernetes deployment

Kubernetes deployments can encounter several challenges and complexities that may lead to operational inefficiencies and increased risks.

Manual configuration management: Kubernetes configurations, such as YAML files describing pods, services, and deployments, are typically managed manually. This manual approach can lead to human errors, configuration drift, and inconsistencies between environments.

Lack of version control: Traditional Kubernetes configurations might not be stored in version control systems like Git, making it difficult to track changes, collaborate effectively, and roll back to previous configurations if issues occur.

Inefficient deployment process: The absence of automation can result in slower and error-prone deployment processes. Manually applying changes to Kubernetes clusters can be time-consuming and increase the likelihood of misconfigurations.

Difficulty in auditing and compliance: Without version control

and proper tracking, it becomes challenging to maintain an audit trail of changes made to the Kubernetes infrastructure.

Limited collaboration: In team environments, collaboration between developers and operations personnel can be challenging when managing Kubernetes configurations manually. Communication gaps and misunderstandings may occur.

Complex rollbacks: Rolling back changes to a previous stable state can be a complex and manual process, potentially leading to downtime and service disruptions.

Inconsistent environments: Manually managing configurations for different environments (e.g., development, staging, production) can result in inconsistencies, leading to unexpected behaviour and making it harder to test and reproduce issues.

Security vulnerabilities: Manually applying changes to Kubernetes may increase the risk of security vulnerabilities, especially if there are deviations from best practices or if proper access controls are not enforced.

DevOps bottlenecks: Without automation, development teams might rely heavily on operations personnel to manage deployments and configurations, leading to DevOps bottlenecks and slower delivery cycles.

Scalability challenges: As Kubernetes clusters and applications grow in complexity, manual management becomes less scalable and may hinder efficient scaling and maintenance of the infrastructure.

Let us look at how to mitigate these challenges and promote streamlined, collaborative, and secure operational workflow that aligns well with modern DevOps practices and containerised environments.

GitOps

GitOps is a modern software development and operations approach that uses version control systems, typically Git, as the single source of truth for managing infrastructure and application configurations. The core idea behind GitOps is to treat the entire system as code and use Git workflows to automate the deployment and management of applications in a containerised environment, such as Kubernetes.

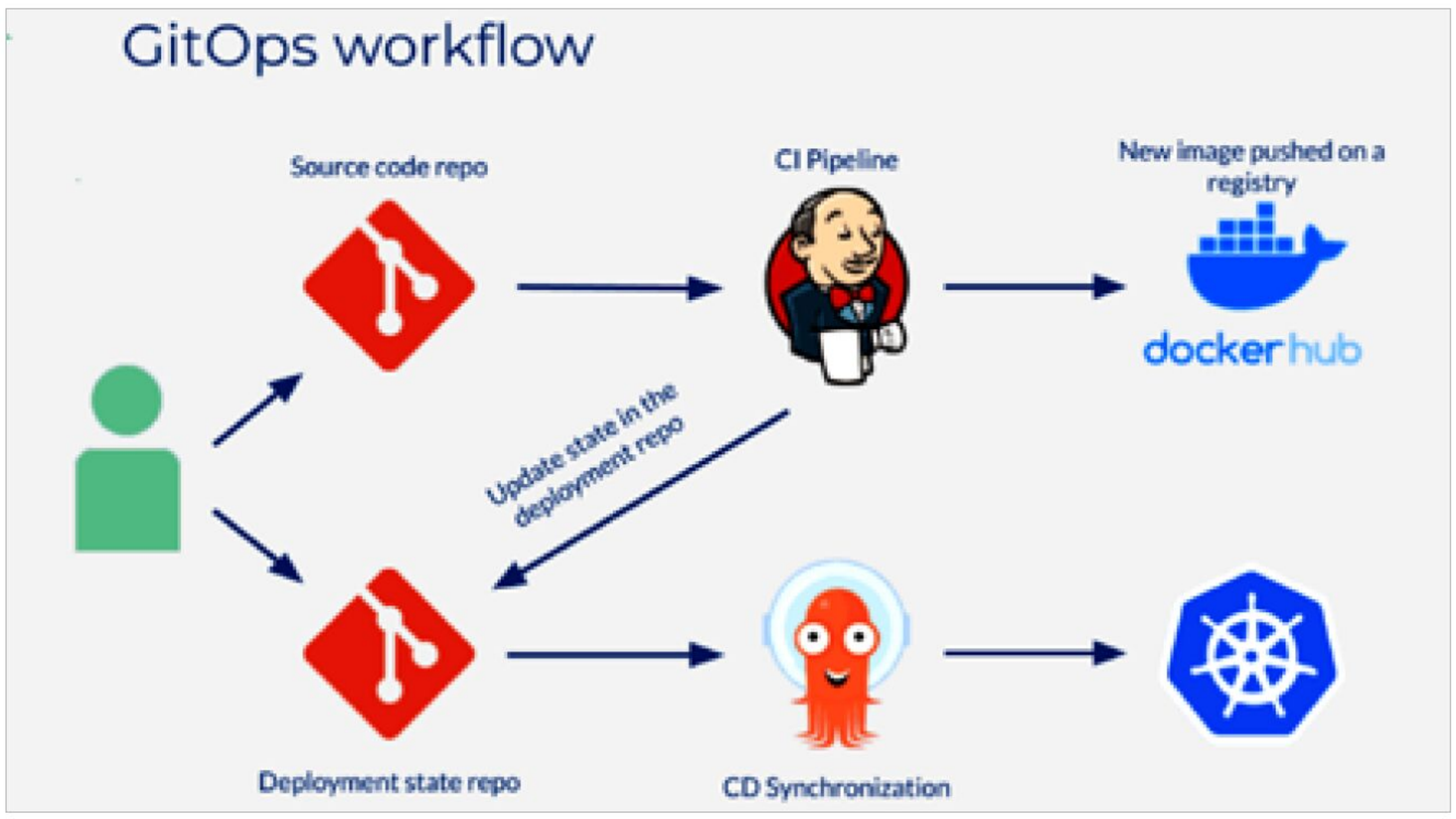


Figure 1: GitOps

Key principles of GitOps

Here we explore how GitOps leverages version control for automated and declarative management, bringing simplicity and efficiency to modern software workflows.

Declarative infrastructure as code: In GitOps, all aspects of infrastructure and application configurations are defined declaratively in code, specifying the desired state of your system rather than writing imperative commands to achieve that state.

Git as source of truth: Git is used as the central repository to store the desired state of the system. Any changes to the infrastructure or application configurations must be made through commits to the Git repository.

***Continuous deployment and automation*:** GitOps workflow emphasises continuous deployment and automation. Changes made to the Git repository trigger automated processes and apply those changes to the target environment.

Reconciliation loop: GitOps employs a reconciliation loop to ensure that the actual state of the system matches the desired state defined in the Git repository. If any deviations are detected, the system automatically makes corrections to bring itself back to the desired state.

How GitOps works

The GitOps workflow typically involves these steps for seamless orchestration.

a. ***Desired state in Git:*** Developers define the desired state of the infrastructure and applications in Git repositories as code. This includes Kubernetes manifest files, application configurations, and other related resources.

b. ***Git repository as source of truth:*** The Git repository becomes the source of truth for the entire system. Any changes to the system should be made through Git commits.

c. ***Continuous deployment:*** A GitOps operator or tool (e.g., Flux, Argo CD) watches the Git repository for changes. When changes are detected, the operator applies those changes to the Kubernetes cluster automatically.

d. ***Reconciliation loop:*** The GitOps operator continuously reconciles the actual state of the cluster with the desired state defined in the Git repository. If any drift is

detected, it brings the cluster back to the desired state by applying the necessary changes.

Benefits of GitOps

Embracing GitOps empowers modern teams to achieve greater efficiency and reliability in their software delivery processes.

Simplified operations: GitOps simplifies the deployment and management of applications by using familiar Git workflows, reducing the need for complex manual operations.

Consistency and reproducibility: By managing configurations as code in Git, GitOps ensures that the same configuration used in development is applied to staging and production environments, leading to consistency and reproducibility.

Version control and collaboration: Git provides version control, enabling teams to collaborate effectively, review changes, and maintain a history of system modifications.

Automated auditing and compliance: GitOps enables easier auditing and compliance tracking since all changes are recorded in Git repositories.

GitOps tools

GitOps tools provide continuous delivery and deployment and adhere to the GitOps approach, where the desired state of the infrastructure and applications is version-controlled in Git repositories. These tools continuously monitor the repositories for changes and automatically apply those changes to the target environment, typically a Kubernetes cluster. Though several open source tools like Jenkins-x, Spinnaker, Kustomize and GitLab are available, we will quickly look into the workings of two widely adopted tools — Argo CD and Flux.

Argo CD

Argo CD is an open source continuous delivery tool that provides a powerful and efficient way to manage Kubernetes applications, offering automated continuous delivery and the benefits of the GitOps workflow that runs on top of Kubernetes.

With Argo CD, you can define your application configurations declaratively in Git repositories and have Argo CD automatically reconcile the state of your applications to match the desired state defined in those repositories. It is widely adopted by organisations looking to simplify application deployments, reduce manual interventions, and improve the overall reliability of their Kubernetes environments.

How Argo CD works

Application definition: You define your application specifications, including the deployment manifests, service configurations, and any other required resources, in a Git repository.

Argo CD server: The Argo CD server continuously monitors the Git repository for changes. When a change is detected, the server initiates a reconciliation process.

Reconciliation: During reconciliation, Argo CD compares the desired state in the Git repository with the actual state in the Kubernetes cluster. If there are differences, Argo CD makes the necessary changes to bring the cluster in line with the desired state.

Application deployment: Argo CD deploys and manages your applications using Kubernetes manifests, Helm charts, or other supported application definition formats.

Application synchronisation: Argo CD keeps the applications continuously synchronised with the Git repository. Any changes made directly to the cluster are automatically reverted, ensuring that the desired state remains consistent.

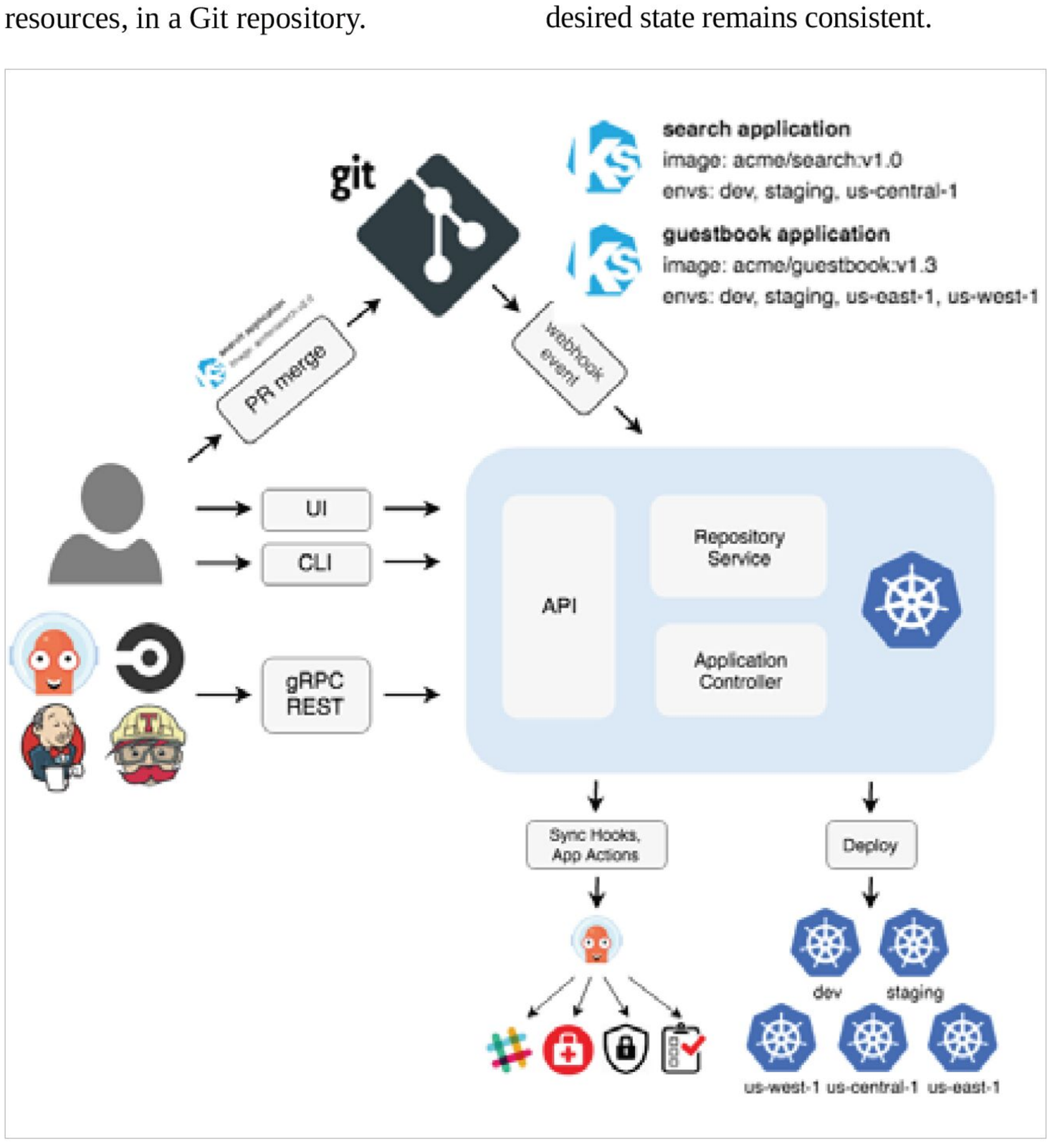


Figure 2: ArgoCD

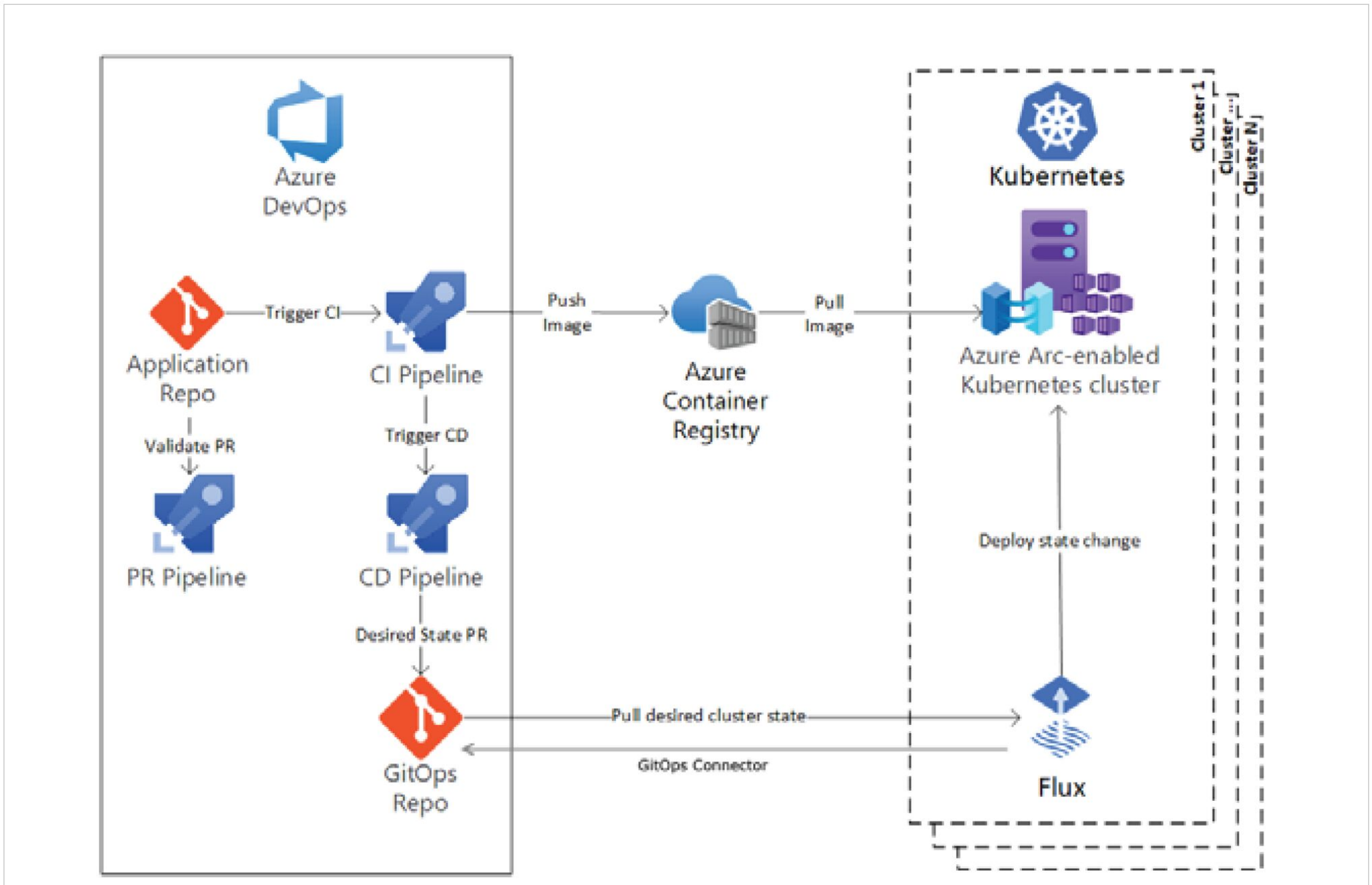


Figure 3: Flux

Flux

Flux is an open source continuous delivery tool that follows the GitOps approach to automate the deployment and management of applications in Kubernetes. Similar to Argo CD, Flux uses Git as the single source of truth for defining the desired state of applications and infrastructure configurations. It continuously monitors Git repositories and automatically synchronises the Kubernetes cluster with the configurations defined in those repositories.

How Flux works

Application definitions: You define the desired state of your applications and infrastructure configurations in a Git repository using Kubernetes manifests or Helm charts.

Flux controller: The Flux controller continuously monitors the Git repository for changes. It checks for new commits and updates to the defined configurations.

Reconciliation: When changes are detected, the Flux controller initiates a reconciliation process. It compares the cluster's actual state with the desired state defined in the Git repository.

Automated deployment: Flux automatically applies the changes to the cluster, deploying and managing applications based on the configurations defined in Git.

Synchronisation: Flux keeps the cluster continuously synchronised with the Git repository. Any changes made directly to the cluster are reverted during the next synchronisation to ensure that the cluster remains in sync with the desired state.

Both Argo CD and Flux provide support for Helm charts, allowing you to deploy Helm releases using the CI/CD pipeline. Additionally, you can set up custom policies in both tools to define the automated deployment behaviour, such as automated vs manual deployments and strategies for updating application versions.

It's essential to secure access to the Kubernetes cluster and the Git repository to prevent unauthorised changes. By integrating Argo CD or Flux with your CI/CD pipeline, you can establish a robust and automated GitOps-based workflow for deploying applications and configurations to your Kubernetes cluster with ease. END

By: Renganathan D.

The author is a Google cloud certified architect at HCL Technologies Ltd.

Disclaimer: This article expresses the views of the author and not of the organisation he works in.

The AI and Cybersecurity Capabilities of GitLab's CI/CD Platform

Here's an in-depth exploration of GitLab's innovative CI/CD platform, which is reshaping how we build software by seamlessly blending AI and cybersecurity capabilities. The focus is on GitLab's AI services and their integration with cloud platforms.



GitLab is a renowned internet-based Git repository administrator that offers an all-encompassing CI/CD platform. Its diverse feature set has fundamentally altered the way in which developers collaborate on software deployments. At its core, GitLab's CI/CD platform serves as an integrated toolchain process, connecting various tools to handle all life cycle activities of software development. This integration streamlines how software modifications are constructed, tested, and deployed. By integrating the CI/CD process into the version control system, GitLab enables teams to maintain an agile development workflow. With faster feedback cycles and consistent deployments, teams can accelerate their software delivery.

Continuous integration (CI) in GitLab

Continuous integration within GitLab involves the frequent integration of code changes into a central hub. GitLab's CI system offers a wide range of built-in features, including:

Pipelines: Using '.gitlab-ci.yml' files, developers may create tailored pipeline configurations that outline the activities required to construct, evaluate, and release applications.

Runners: These runners perform CI/CD functions and can be tailored to individual projects, enabling versatility and extensibility.

Parallel execution: CI/CD pipelines optimise build times and enhance development efficiency through parallel job execution.

Continuous deployment (CD) in GitLab

Production environments receive the automatic release of code changes following successful CI testing. GitLab's CD capabilities include:

Environment management: GitLab streamlines the administration of diverse ecosystems, including development, testing, and deployment settings, thus enabling maintenance of standardised setups.

Auto DevOps: By employing best practices, GitLab's Auto DevOps function sets up CI/CD pipelines swiftly and with minimal effort.

Paradigm shift towards Auto DevOps

Auto DevOps boasts an AI-assisted functionality that scrutinises the codebase and establishes automatic CI/CD pipeline configurations. By integrating past data and machine learning models, Auto DevOps streamlines pipeline phases. This encompasses choosing the right runner types and tuning parameters depending on the project's properties. Individualised CI/CD pipeline configurations are established using this method, eliminating the need for manual tweaking.

The Auto Remediation tool is intended to automatically repair failed CI/CD projects. Using AI insights, Auto Remediation provides customised advice or job retry suggestions, saving developers time and effort. This feature reduces the manual intervention required to address common build or test failures, promoting faster delivery of code changes.

GitLab's platform offers extensive cybersecurity options for CI/CD pipelines. It prioritises cybersecurity-enabled features by incorporating robust security measures into its CI/CD platform, allowing developers to detect and mitigate security risks during development.

SAST vs DAST testing in GitLab AI services

The purpose of a static application security testing (SAST) tool is to analyse the source code of software applications to identify potential security vulnerabilities and coding errors. SAST is a type of white-box testing — it examines the internal structure and logic of the code without executing the application. The primary goal of SAST is to proactively detect security flaws early in the software development life cycle, allowing developers to address and fix these issues before the application is deployed or released to production. GitLab's SAST tool performs a detailed review of the source code, locating security weaknesses like code injection, incorrect authentication, and XSS threats. SAST integration within the CI/CD pipeline helps ensure a secure software release by identifying and addressing security concerns earlier in development.

Dynamic application security testing (DAST) is a type of security testing that assesses web applications and APIs while they are running. Unlike SAST, which examines the source code of an application, DAST operates from the outside-in, simulating real-world attacks by interacting with the application through its front-end, just like an external user would. The running state of web applications is checked by GitLab's DAST security feature in their CI/CD platform. Using DAST to scan deployed apps reveals potential flaws that may escape detection through other methods. An anticipatory security tactic offers robust defence against prospective dangers. There are two forms of DAST testing in GitLab, which use AI features to automate test management.

Dependency scanning: Automated dependency scanning in GitLab identifies recognised security weaknesses among project dependencies like libraries and packages. Dependency scanning integration within the pipeline minimises the chance of threats exploitation by proactively identifying and flagging obsolete or unsafe components.

Container scanning: Containerisation being more widely adopted has led GitLab to include container scanning in its CI/CD workflow. The feature flags any security concerns found within container images, including antiquated packages or an inadequate setup. Safeguarding containers strengthens comprehensive application security during deployment.

Shifting from DevOps adoption to DevSecOps adoption

DevOps is termed as a 'cultural movement' that emphasises the collaboration and communication between development, operations, and security teams. The goal of DevOps is to shorten the software development life cycle and improve the quality of software by automating the delivery process.

DevSecOps is an extension of DevOps that brings security into the fold. It ensures that security is considered at every stage of the software development life cycle, from the initial design phase to the final deployment.

Here are some of the key differences between DevOps adoption and DevSecOps adoption:

- DevOps focuses on speed and efficiency, while DevSecOps focuses on security. DevOps teams are often focused on getting code out the door quickly, while DevSecOps teams are focused on making sure that code is secure before it is deployed.
- DevOps teams are often siloed, while DevSecOps teams are more collaborative. DevOps teams are often made up of developers and operations engineers, while DevSecOps teams also include security engineers. This collaboration helps to ensure that security is considered at

every stage of the software development life cycle.

- DevOps teams use a variety of tools to automate the software delivery process, while DevSecOps teams use more specialised tools to scan code for security vulnerabilities and to automate security testing.

Here are some of the benefits of shifting from DevOps adoption to DevSecOps adoption.

Improved security: DevSecOps can help organisations identify and fix security vulnerabilities earlier in the software development process, which can help to reduce the risk of data breaches and other security incidents.

Improved compliance: DevSecOps can help organisations comply with industry regulations, such as PCI DSS and HIPAA.

Increased efficiency: DevSecOps can help to streamline the software delivery process, which can lead to faster time to market and reduced costs.

Improved developer productivity: It can help developers to be more productive by reducing the time they spend on security tasks.

Here are some steps you can take to shift from DevOps adoption to DevSecOps adoption.

Educate the team about DevSecOps: Make sure that everyone on the team understands the principles of DevSecOps and why it is important.

Identify security risks: Assess the current security posture and identify the areas where you are most vulnerable.

Implement security controls: Put in place security controls to mitigate the risks being identified.

Automate security testing: Use automated tools to scan the code for security vulnerabilities and to automate security testing.

Monitor your security posture: Continuously monitor the security posture to ensure that you are staying ahead of the latest threats.

Shifting from DevOps adoption to DevSecOps is a journey, not a destination. It takes time and effort to fully implement DevSecOps principles. However, the benefits of DevSecOps can be significant, including improved security, compliance, efficiency, and developer productivity.

Enabling GitLab AI services for the cloud

The following are the various GitLab AI services for a cloud platform.

- ***Code intelligence:*** GitLab AI can be used to provide code intelligence, which can help developers to write more secure and efficient code. For example, GitLab AI can help developers to identify potential security vulnerabilities, code smells, and inefficient code patterns.
- ***Code review:*** GitLab AI can be used to automate code review, which can help to improve the quality of code. For example, it can be used to identify potential security vulnerabilities and code smells.
- ***Security testing:*** GitLab AI can be used to automate security testing, which can help to identify and fix security vulnerabilities. For example, it can be used to scan code for known security vulnerabilities.
- ***Compliance reporting:*** GitLab AI can be used to generate compliance reports, which can help organisations to demonstrate compliance with industry regulations. For example, GitLab AI can be used to generate a report of all security vulnerabilities that have been identified in code.
- ***AI-assisted code review:*** This feature uses machine learning to identify potential security vulnerabilities and code smells in your code. It can also suggest improvements to your code, such as making it more efficient or easier to read.
- ***AI-assisted security testing:*** This feature uses machine learning to scan your code for known security vulnerabilities. It can also identify potential security vulnerabilities that are not yet known to the public.
- ***AI-assisted compliance reporting:*** This feature uses machine learning to generate compliance reports that demonstrate your organisation's compliance with industry regulations. It can also identify potential compliance gaps that need to be addressed.

Follow these steps to enable Gitlab AI services in a cloud platform.

1. To enable the AI/ML powered features setting, go to your GitLab instance and click on the 'Settings' icon. Then, click on 'General' and scroll down to the AI/ML powered features section. Check the box next to 'Enable AI/ML powered features' and click on 'Save changes'.
2. To create a Google Cloud service account, go to the Google Cloud platform console *https://console.cloud.google.com/* and click on the 'IAM & Admin' tab. Then, click on 'Service accounts' and on the 'Create service account' buttons. In the service account name field, enter a name for your service account. In the role field, select the 'AI Platform Admin' role. Click on the 'Create' button.
3. Once your service account has been created, click on the 'Download JSON key file' button. This will download a JSON file containing the credentials for your service account.
4. To create an environment variable in your GitLab project, go to your project's 'Settings' page and click on the 'Environment variables' tab. Click on the 'Add variable' button and enter these values.

- *Name:* GITLAB_CLOUD_SERVICE_ACCOUNT_CREDENTIALS
- *Value:* The contents of the JSON key file for your Google Cloud service account

5. To restart your GitLab Runner, you can use this command:

```
sudo gitlab-runner restart
```

On completion of these five steps, GitLab AI services will be enabled on the cloud platform. And GitLab AI features, such as code intelligence and code review, can be used to improve the quality and security of the code.

Benefits of GitLab AI services

GitLab AI services offer several benefits that can enhance the development and deployment of AI projects. Some of the key benefits are listed here.

- ***Integrated AI capabilities:*** GitLab AI services provide built-in AI capabilities, allowing developers to seamlessly incorporate AI functionality into their projects. This integration eliminates the need for separate AI platforms or tools, streamlining the development process.
- ***Version control for AI models:*** GitLab's version control system enables efficient management of AI models. Developers can track changes, collaborate with team members, and easily revert to previous versions if needed. This ensures better control and reproducibility of AI models throughout their life cycle.
- ***Continuous integration and deployment (CI/CD):*** GitLab AI services support CI/CD pipelines, enabling automated testing, building, and deployment of AI models. This automation reduces manual effort, speeds up the development cycle, and ensures consistent and reliable deployment of AI applications.
- ***Collaboration and knowledge sharing:*** GitLab provides a collaborative environment for AI development teams. Developers can work together, share code, and review each other's work, fostering collaboration and knowledge sharing. This accelerates learning, improves code quality, and promotes best practices in AI development.
- ***Scalability and performance:*** GitLab AI services are designed to handle large-scale AI projects. With features like distributed computing and parallel processing, developers can leverage the power of multiple machines to train and deploy AI models efficiently. This scalability ensures optimal performance even for resource-intensive AI workloads.
- ***Security and compliance:*** GitLab prioritises security and compliance, providing features like access controls, authentication, and encryption. This ensures that AI models and data are protected throughout the development and deployment process, meeting industry standards and regulatory requirements.
- ***Extensibility and customisation:*** GitLab AI services are highly extensible, allowing developers to integrate with other AI frameworks, libraries, or tools. This flexibility enables customisation and integration with existing AI ecosystems, empowering developers to leverage their preferred tools and workflows.

Overall, GitLab AI services offer a comprehensive platform for AI development, combining version control, collaboration, automation, and scalability. These benefits contribute to faster development cycles, improved code quality, and efficient deployment of AI applications.

GitLab's innovative platform has disrupted software creation by seamlessly blending AI and cybersecurity capabilities. AI-based functionalities maximise workflow effectiveness, improve output, and provide specialised CI/CD pipelines for every project. These include Auto DevOps and Auto Remediation. GitLab's security features, including SAST, DAST, dependency scanning, and container scanning, strengthen the development process. These features ensure that software projects are reliable and secure from inception to deployment. Development teams can efficiently harness these sophisticated functions to construct premium software and then successfully deploy it into production settings. END

References

- *https://devops.com/gitlab-adds-more-ai-and-cybersecurity-capabilities-to-ci-cd-platform/*
- *https://about.gitlab.com/blog/2023/04/24/ai-ml-in-devsecops-series/*
- *https://about.gitlab.com/solutions/security-compliance/*
- *https://docs.gitlab.com/ee/user/ai_features.html*

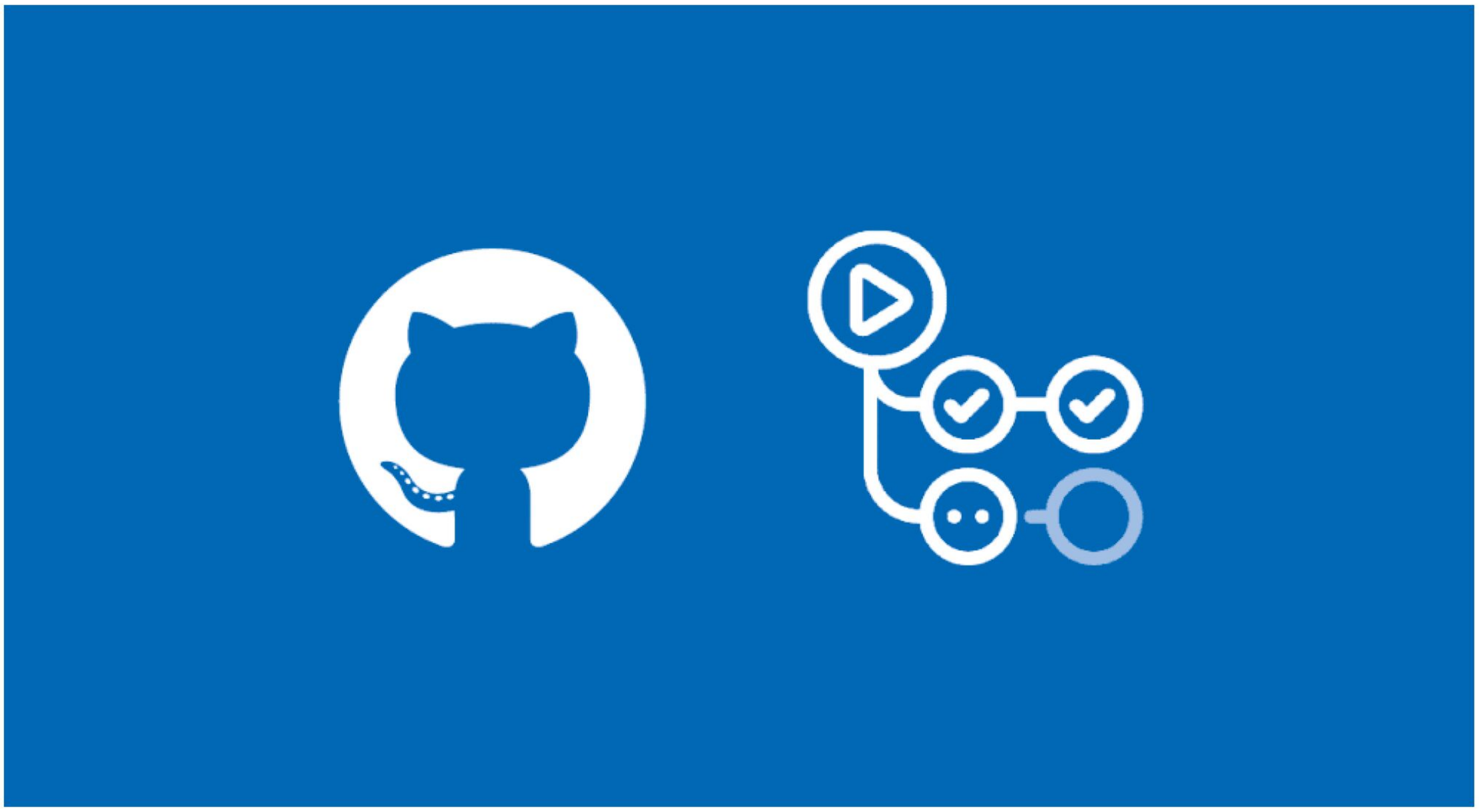
By: Dr Magesh Kasthuri and Dr Anand Nayyar

Dr Magesh Kasthuri is a senior distinguished member of the technical staff and principal consultant at Wipro Ltd. This article expresses his views and not that of Wipro.

Dr Anand Nayyar is a PhD in wireless sensor networks and swarm intelligence. He works at Duy Tan University, Vietnam, and loves to explore open source technologies, IoT, cloud computing, deep learning and cyber security.

Continuous Integration with GitHub Actions

Here are a few insights into the core principles of DevOps and some practical guidance on using GitHub Actions.



DevOps isn't the name of a team or a job title or a particular technology. Instead, it's a set of processes, ideas, and techniques that combines people, cultural practices, and technologies. The goal is to make software delivery vastly more efficient. DevOps helps teams ship high-quality products faster by reducing the friction between writing, testing, and deploying code.

What is the structure of a DevOps model? Well, it does not work on a particular model; rather it is an approach to building software to enhance the entire development cycle. DevOps fundamentally seeks to transform organisations by bringing traditionally siloed teams together across every part of the software development life cycle.

Key aspects of DevOps

***Collaboration*:** DevOps needs close cooperation between operations and development for a productive environment.

***Version control*:** Version control is an integral part of DevOps. A version control system is designed to automatically record file changes and preserve records of previous file versions.

***Automation*:** It establishes feedback loops between those in charge of maintaining the underlying infrastructure and those in charge of developing the core software.

***Incremental releases*:** Successful DevOps practices rely on incremental releases, which are characterised by the speedy delivery of feature additions and updates. Incremental releases enable development teams to swiftly respond to business needs.

***Orchestration*:** Orchestration aids in configuring, managing, and coordinating the infrastructure necessary for an application to function properly.

***Pipeline*:** The term 'pipeline' frequently arises in discussions on DevOps. A DevOps pipeline sequences various stages of the software development life cycle like compiling and analysing the code, unit testing, building the image, containerising the image, and more.

GitHub Actions

GitHub Actions is a popular tool used in DevOps. It helps in implementing continuous integration and continuous deployment (CI/CD) besides other automation.

CI aids teams in promptly identifying and resolving issues that arise early in the development process. For example, every time a developer submits a code change to the GitHub repository, a CI process builds and tests the application immediately. Such a CI process can be built using GitHub Actions.

Components of GitHub Actions

Let's peek under the hood of GitHub Actions and discover how its various parts work together.

***Workflow*:** A workflow is an automation pipeline for a given GitHub repository. It is defined in a YAML file under the '.github/workflows' folder. A workflow consists of one or more jobs triggered by repository events.

***Event*:** An event signifies a change in the repository's status. For example, an event occurs when someone pushes code into a repository or makes a pull request, etc. A workflow can be configured to run against one or more such events. Moreover, a successful workflow can also trigger another workflow.

***Job*:** A job is a sequence of steps to be executed such as scripts or other pre-defined actions. Multiple jobs can be defined in a workflow. The jobs in a workflow run independently in parallel unless they depend on each other. A job succeeds only if all the steps in it are executed without errors. Similarly, a workflow succeeds only if all the jobs defined in it get executed successfully.

***Action*:** To reduce the complexity of defining workflows GitHub offers pre-defined actions. For example, there are actions available to set up a Node environment, or a Java environment. The large community of GitHub Actions also offers several useful actions through the marketplace. All you need to do is correctly configure them. You can also define your own custom actions based on specific requirements.

***Runners*:** We need servers to execute the jobs of workflows. These servers are called runners. By default, the jobs run on GitHub-hosted runners. They are available on Linux, macOS, and Windows platforms. We can also configure our own runners for special purposes such as safety, security, capacity, etc.

A use case for GitHub Actions

Let us consider the case of creating a Docker container image from a Java project managed by Maven. As you know, Maven handles dependency management apart from compilation, building, testing, and packaging. A Docker container is a self-contained image that can run as it is on Docker runtime.

Here we will automate a CI process to build and push the Docker image whenever someone pushes the code into the repository.

Set up a workflow: To create a workflow, follow these steps:

- Log in to GitHub, visit the repository of the code, click the 'Actions' menu item and press the 'New workflow' button.
- Search for 'Simple workflow' and click on 'Configure'. It creates a YAML file and displays the default workflow in the editor.
- Replace the code in the YAML with the following code snippet:

```
name:  CI Workflow
on:
  pull_request:
    types: [opened, reopened]
    branches:
      - main

  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest
    permissions:
      contents: read
      packages: write
    steps:
      - name: Checkout Repository
        uses: actions/checkout@v2
      - name: Set up JDK 8
        uses: actions/setup-java@v2
        with:
          java-version: '8'
          distribution: 'temurin'
      - name: Build with Maven
        run: |
          mvn -B package --file pom.xml
      - name: Build and Push Docker
Image
        uses: mr-smithers-excellent/
docker-build-push@v5
        with:
          image: glarimy/glarimy-
assessements-api
          tags: latest
          registry: docker.io
          dockerfile: Dockerfile
          username: ${{SECRETS.DOCKER_
ID}}
          password: ${{SECRETS.DOCKER_
PWD}}
```

The name of the workflow is CI Workflow. The code inside 'on' defines when the workflow will trigger. This workflow is configured to be triggered when someone makes a pull request to the main branch, or someone pushes a file in the main branch.

In 'jobs' there is only one job named 'build'. To run a job, you need a runner. The runner can be a GitHub-hosted runner or a self-hosted runner. In the code snippet, we used the GitHub-hosted runner ubuntu-latest. The permissions to the job come next.

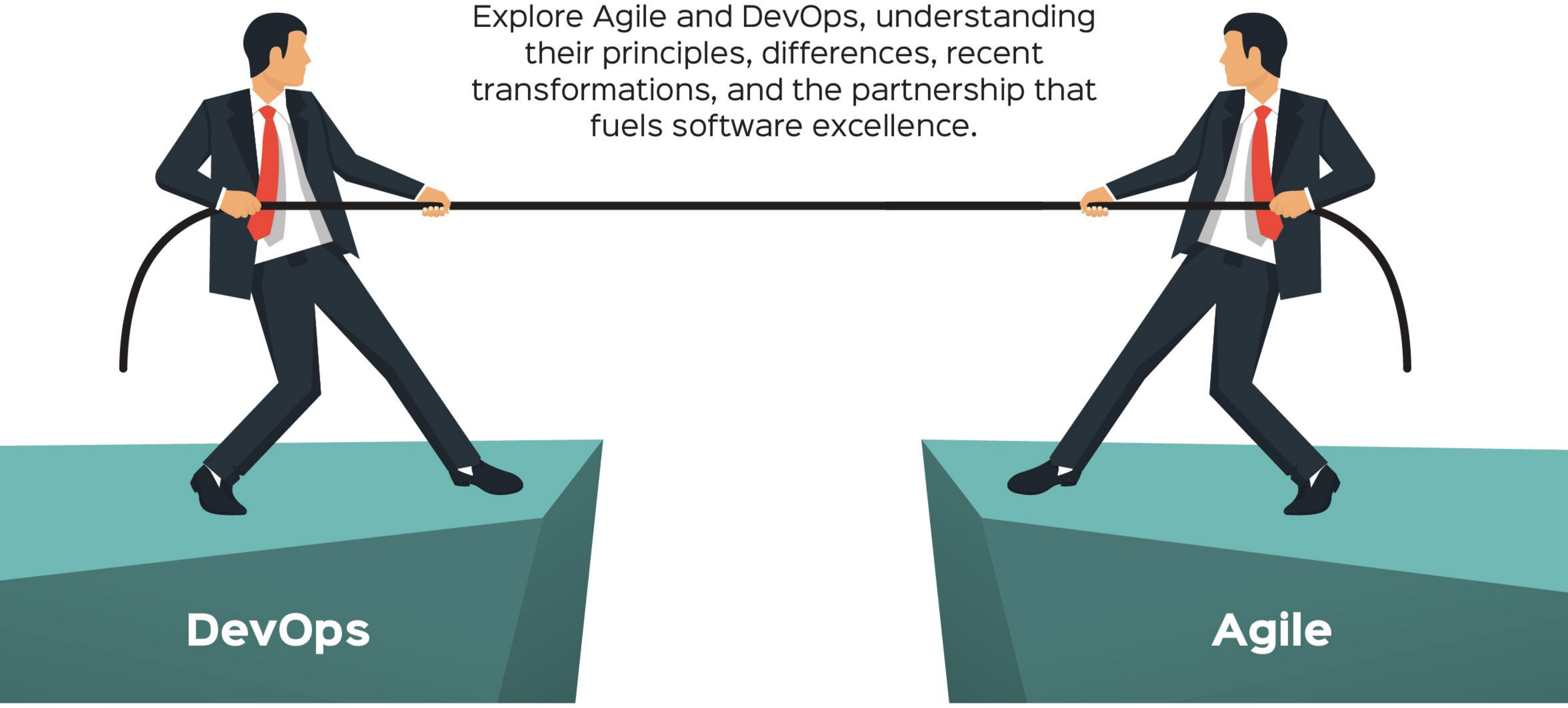
Let's now define the steps to be followed by the job for CI workflow. In our case, there are four steps.

Step 1: Check out the repository on the runner. We use a pre-defined action for this purpose.

Continued to page...51

DevOps Vs Agile:
What You Should Know

Explore Agile and DevOps, understanding their principles, differences, recent transformations, and the partnership that fuels software excellence.



In the fast-paced and ever-evolving world of software development, two methodologies have emerged as game changers, empowering teams to deliver high quality products efficiently and swiftly. DevOps and Agile are two distinct approaches that have gained significant popularity in recent years, each with its own set of principles and advantages. While they share some similarities, they address different aspects of the development process and serve unique purposes.

In this article, we will delve into the world of DevOps and Agile, highlighting their key differences, core principles, and how they can complement each other. By understanding these methodologies, you'll be better equipped to make informed decisions for your team and project, optimising your software development processes and achieving remarkable results. So, buckle up, as we embark on this journey of discovery into the realms of DevOps vs Agile.

Knowing Agile

Agile is the go-to approach for iterative and incremental development. It's all about collaboration, customer feedback, and adaptability to changes. Agile teams work in short development cycles called 'sprints', which usually last around two to four weeks. And Scrum, one of the most popular Agile frameworks, is all about that!

Here are Agile's core values, as laid out in the Agile Manifesto:

- People and interactions are more vital than processes and tools.
- Working software beats comprehensive documentation any day.
- Customer collaboration trumps contract negotiation.
- Being able to adapt to change is way better than following a rigid plan.

Knowing DevOps

DevOps is not a development methodology; rather, it's a set of practices that foster collaboration and communication between development and operations teams. It's all about breaking down barriers and creating a culture of shared responsibility throughout the entire software delivery process.

The key principles of DevOps are as follows.

- ***Collaboration is the name of the game:*** DevOps puts collaboration front and centre. It's all about smashing those silos between development, operations, and other teams involved in the software life cycle.

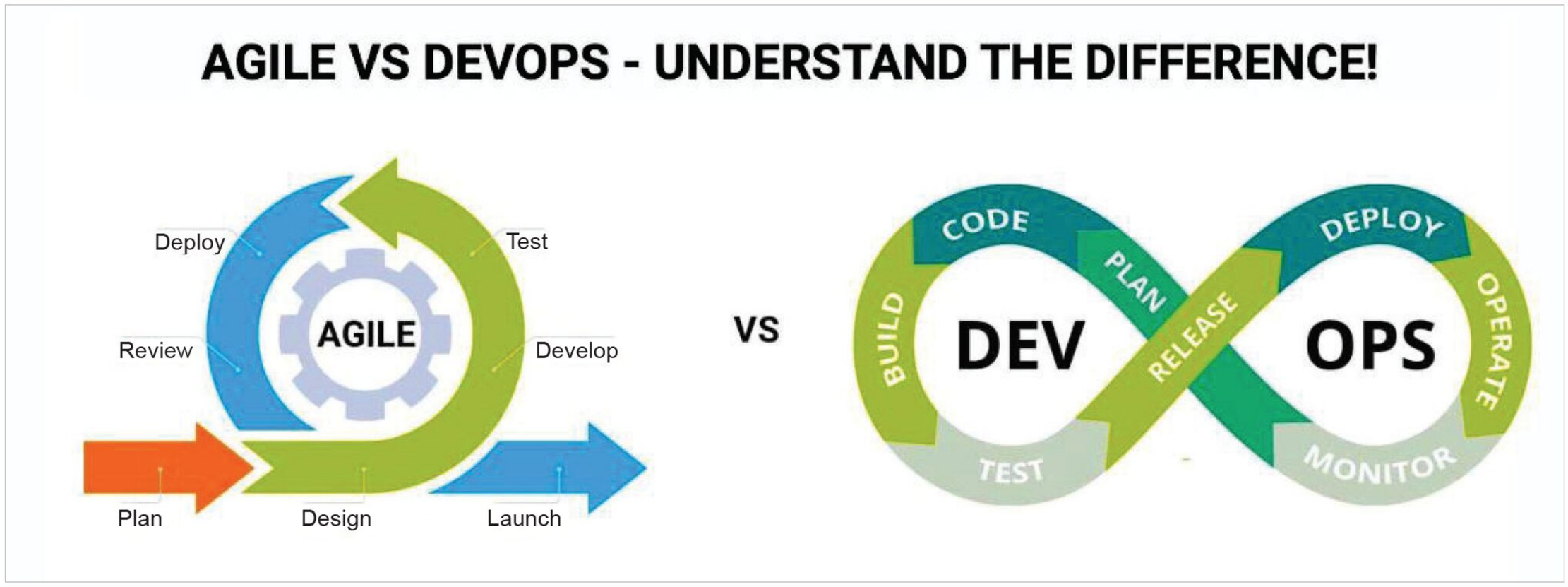


Figure 1: The differences between Agile and DevOps

- ***Automate everything:*** Imagine automating tasks like building, testing, and deploying software. Cool, right? That's what DevOps is all about. It speeds up the delivery process and makes it more reliable.
- ***Say 'hello' to CI/CD:*** Continuous integration and continuous deployment (CI/CD) are DevOps' BFFs. They help integrate code changes frequently and automate the whole deployment process.
- ***Feedback is golden:*** DevOps doesn't shy away from feedback. Monitoring tools are used to gather performance and user experience insights, making room for continuous improvement.

Differences between Agile and DevOps

Agile and DevOps may seem like two peas in a pod, but they have their differences.

1. ***Scope***
- Agile focuses on the development phase, building and prioritising features.
- DevOps covers the entire software delivery pipeline, from development to testing, deployment, and operations.

2. ***Focus***
- Agile aims to deliver the most valuable features to customers regularly within short sprints.
- DevOps focuses on collaboration and automation, ensuring speedy and reliable deployments.

3. ***Team structure***
- Agile thrives on cross-functional teams working together in sprints.
- DevOps breaks down walls between Dev (development) and Ops (operations), creating a more integrated and collaborative team.

4. ***Timeframe***
- Agile operates in fixed-length iterations (sprints) of one to four weeks.
- DevOps goes for continuous delivery, slashing the time between code changes and deployment.

Complementary relationship

DevOps and Agile, when combined, create a powerful synergy that enhances software development in various ways.

Faster time-to-market: Agile's iterative development, combined with DevOps' automated deployment, ensures rapid and continuous delivery, reducing time-to-market.

Continuous feedback loop: Agile's customer collaboration is amplified by DevOps' monitoring tools, providing real-time feedback for improvements.

Enhanced quality: Agile's focus on testing and DevOps' automation result in higher software quality and reliability.

Efficient collaboration: DevOps encourages a culture of collaboration among teams, reinforcing Agile's emphasis on cross-functional cooperation.

Adaptability: Agile's iterative approach and DevOps' automation allow teams to adapt quickly to changing requirements and market demands.

Recent changes in Agile and DevOps

In response to evolving industry needs, Agile and DevOps have embraced transformative changes.

- ***SecOps integration:*** With an increasing focus on security, DevOps has evolved to include SecOps practices. Security is integrated into the entire development life cycle, ensuring the early detection and prevention of vulnerabilities.
- ***AI and automation:*** DevOps and Agile are leveraging artificial intelligence and machine learning to further automate processes, optimise workflows, and enhance decision-making.
- ***Site reliability engineering (SRE):*** SRE, an approach within DevOps, focuses on maintaining the reliability of applications in production. It emphasises resilience, monitoring, and rapid incident response.

The verdict: A powerful duo

Agile and DevOps are a match made in software heaven! While Agile lays the foundation with customer-centricity and iterative development, DevOps steps in with automation and continuous delivery to turbo-charge the process. Together, they form a powerful duo that can take your software development to new heights. This complementary relationship boosts productivity, customer satisfaction, and overall software quality, making them essential methodologies for modern development teams.

Remember, there's no one-size-fits-all solution when it comes to choosing between Agile and DevOps. It depends on your team's specific needs and your organisation's goals. Some teams may adopt Agile first and then introduce DevOps practices gradually, while others might implement both methodologies simultaneously.

In conclusion, DevOps and Agile are two remarkable methodologies that have transformed the software development landscape. Agile brings customer focus, adaptability, and collaboration, while DevOps contributes automation, speed, and reliability. By understanding their unique benefits and how they complement each other, you can make an informed decision on which approach best suits your team and project.

So, whether you're on Team Agile, Team DevOps, or the perfect combination of both, get ready to deliver outstanding software and delight your users. END

References

- *https://devops.com/*
- *https://scaledagileframework.com*

By: Neetesh Mehrotra

The author works at Ernst & Young as a senior. His areas of interest are Java development and automation testing.

Continued from page...48

Step 2: Set up JDK 8 from Temurin distribution on the runner. This step also uses a pre-defined action.

Step 3: Build the code with Maven. Here we are using the 'mvn package' command to build the project and create a distributable package. This step generates a JAR on the runner. The name of JAR depends on the way *pom.xml* is configured.

Step 4: Build and push a Docker image. This is also accomplished as a set of commands specific to Docker. The Docker image is built as per the Dockerfile definition available in the repository. Exposing sensitive information like Docker credentials is not the best practice. To overcome this, GitHub allows us to define secrets. The secrets are stored in an encrypted form.

That's all! Whenever a new code is pushed to the repository, the workflow runs and generates a Docker container image if the code passes through the build process successfully.

DevOps is a collaborative approach in software development and IT operations. It is an ever-evolving approach, introducing new practices and tools to streamline development and delivery processes. END

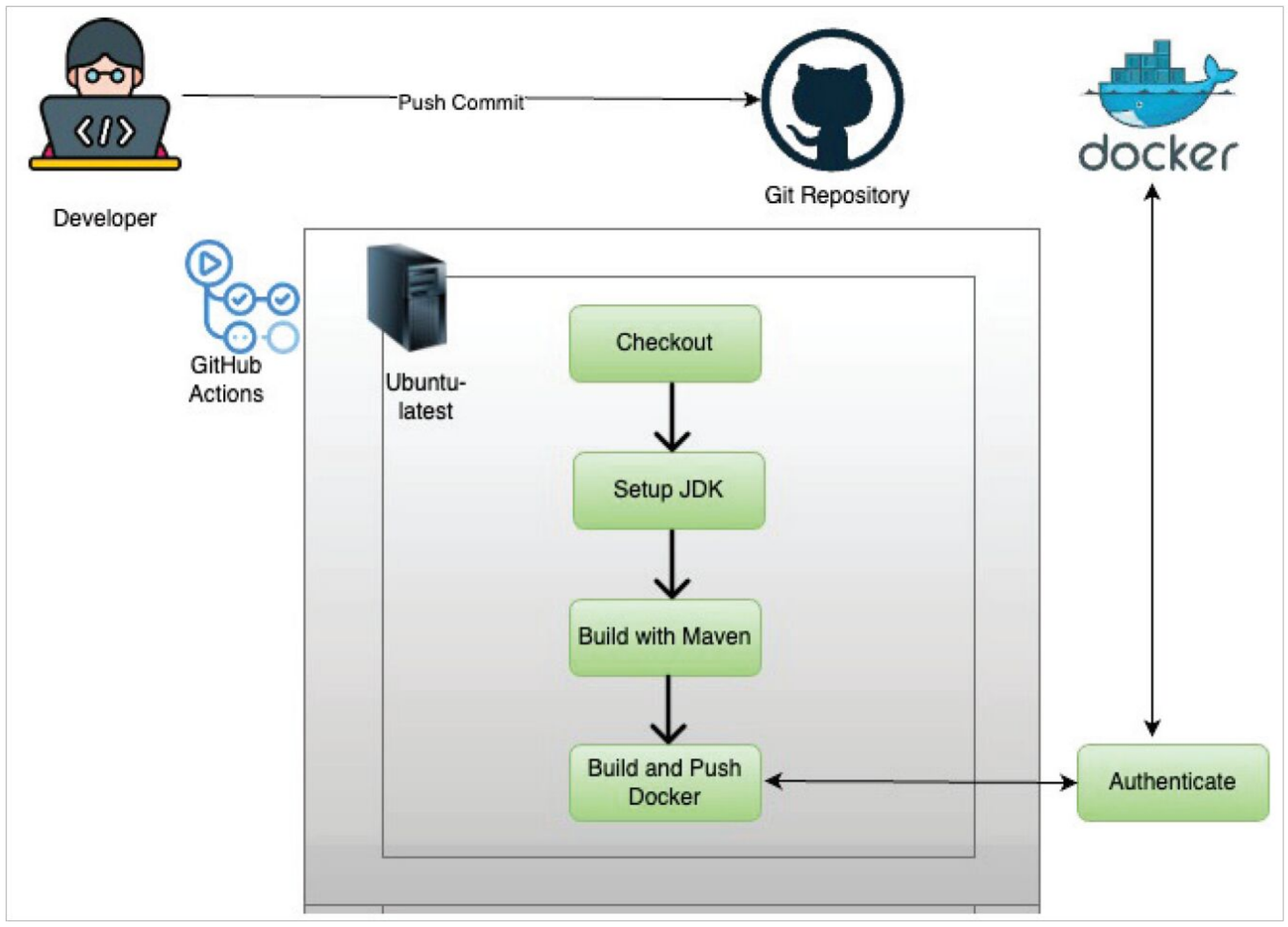


Figure 1: Use case

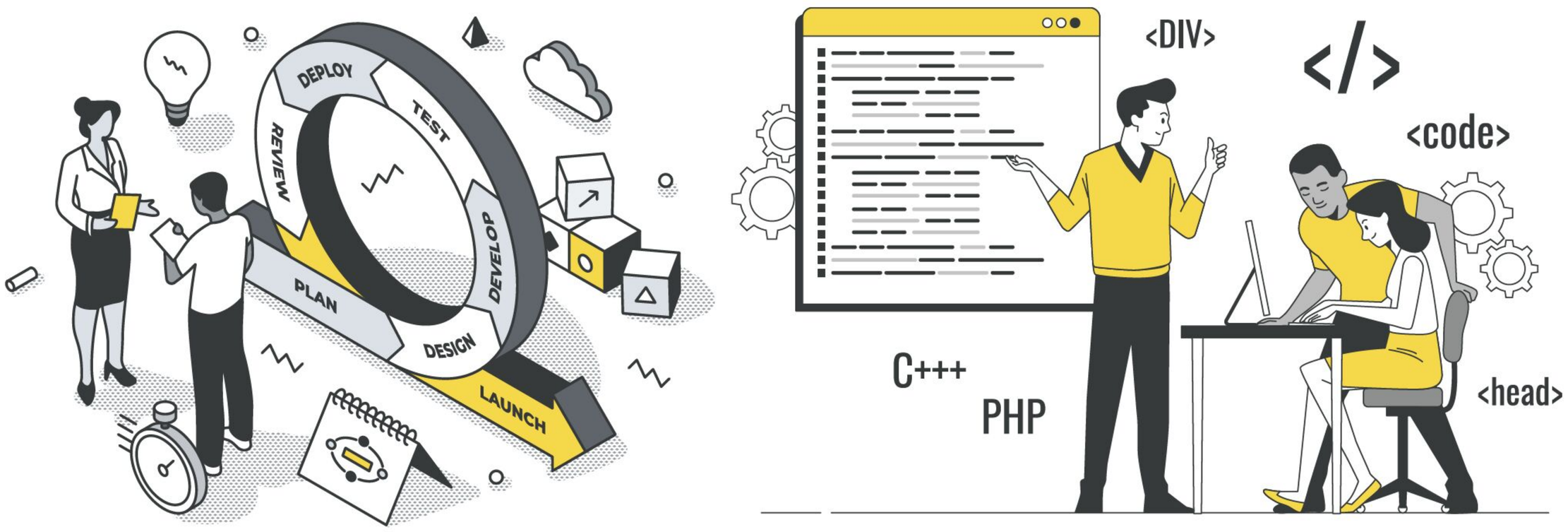
By: Manojeet Dey and Dheerajnadh Jammalamadugu

Manojeet Dey is a computer science enthusiast committed to lifelong learning. He has a strong foundation in the fundamentals of computer science, such as data structures, algorithms, and object-oriented programming. He is proficient in Java and C++.

Dheerajnadh Jammalamadugu is passionate about computer science and is well-versed in the Java and Python programming languages. He has delved into various machine learning algorithms and successfully constructed predictive models.

How DevOps Differs from Traditional IT, and Which One is Better

DevOps enhances collaboration, while traditional IT upholds stability. Knowing when to automate and when to rely on stability is key. Organisations must choose their strategy wisely for optimal outcomes in varied industries. A balance of both DevOps and traditional IT generally works well.



In the rapidly evolving landscape of information technology (IT), two distinct approaches to software development and IT operations have emerged: DevOps and traditional IT. These approaches differ in their methodologies, emphasis, and outcomes. In this article, we will explore the key characteristics of DevOps and traditional IT, and compare how they apply to two different organisations – a stock broker and a healthcare organisation. By understanding how each approach aligns with the unique needs and objectives of these organisations, we can gain insights into which method might be more suitable for their specific contexts.

Overview of DevOps and traditional IT

DevOps is a collaborative approach that encourages seamless communication and cooperation between software developers and IT operations teams. Its main focus is on breaking down silos and promoting a culture of continuous learning and improvement. By integrating development and operations, DevOps aims to accelerate software delivery, enhance product quality, and increase customer satisfaction.

Traditional IT follows a more rigid and compartmentalised structure, with distinct development and operations teams. Communication between these teams may be limited, leading to potential delays and misunderstandings. Traditional IT emphasises stability, control, and well-established processes to maintain reliable IT services and software.

DevOps for a stockbroker organisation

DevOps' collaborative approach and adaptability align perfectly with the fast-paced demands of stock brokers.

Collaboration and communication: In a stockbroker organisation, DevOps facilitates seamless communication between the development team, responsible for building trading platforms and applications, and the operations team, responsible for maintaining

the trading infrastructure. This collaboration ensures that both teams understand the unique requirements of the financial industry and work together to deliver stable and efficient trading solutions.

Emphasis on automation: Automation in a stockbroker organisation expedites the deployment of software updates, allowing for faster response times to market fluctuations. Automated testing also ensures that trading applications are reliable and secure, which are critical factors in highly competitive and time-sensitive financial markets.

Continuous integration/ continuous deployment (CI/CD): With DevOps, the stockbroker organisation can implement continuous integration to integrate code changes frequently, ensuring a consistent and stable codebase. Continuous deployment enables rapid release cycles for new features and improvements, providing traders with the latest tools and functionalities.

Customer-centric approach: DevOps encourages gathering feedback from traders to understand their needs better and deliver solutions tailored to their requirements. A customer-centric approach helps the stockbroker organisation stay ahead of competitors and offer exceptional trading experiences.

Flexibility and agility: In the fast-paced world of finance, DevOps' flexibility and agility are invaluable. The ability to respond quickly to market changes and emerging trends gives the stockbroker organisation a competitive edge and helps them adapt to dynamic market conditions.

Learning from failures: In the stockbroker context, learning from failures is crucial to identify potential risks and vulnerabilities in trading systems. DevOps' culture of learning and continuous improvement leads to robust risk management strategies and more resilient trading platforms.

Traditional IT for a healthcare organisation

Traditional IT may shine through in the critical healthcare sector.

Silos between development and operations: In a healthcare organisation, traditional IT may maintain separate development and operations teams. Developers focus on creating healthcare software applications, while operations teams handle infrastructure and data management. This structure can lead to potential communication gaps between the teams, affecting the seamless integration of healthcare solutions.

Reliance on manual processes: Traditional IT practices may involve manual testing and deployment of healthcare software. While this approach prioritises stability, it may result in longer deployment cycles and delayed access to critical healthcare applications.

Infrequent releases: Healthcare organisations often prioritise stability and regulatory compliance. As a result, they may prefer infrequent software releases that bundle multiple changes together. This approach may limit the agility of the organisation to respond quickly to emerging healthcare needs.

Stability-centric approach: In the healthcare industry, stability is crucial to ensure patient safety and regulatory compliance. Traditional IT's focus on maintaining reliable systems aligns with the healthcare organisation's objective of providing secure and dependable services.

Formal change management: Healthcare organisations may adopt a formal change management process to ensure that any software updates or changes adhere to strict regulatory requirements. This approach helps mitigate risks associated with software changes in a highly sensitive environment.

Comparing DevOps and traditional IT in stockbroker and healthcare organisations

In the stockbroker organisation, DevOps' emphasis on collaboration, automation, and rapid release cycles aligns well with the dynamic nature of financial markets. The customer-centric approach enables the organisation to deliver trading solutions that meet the specific needs of traders. The flexibility and learning from failures help the stockbroker organisation adapt to evolving market conditions and regulatory changes.

In contrast, the healthcare organisation may prioritise traditional IT's stability-centric approach to maintain reliable healthcare systems. The formal change management process helps ensure compliance with stringent healthcare regulations, while the reliance on manual processes may reduce the risk of unintended disruptions in critical healthcare applications.

Which approach is better?

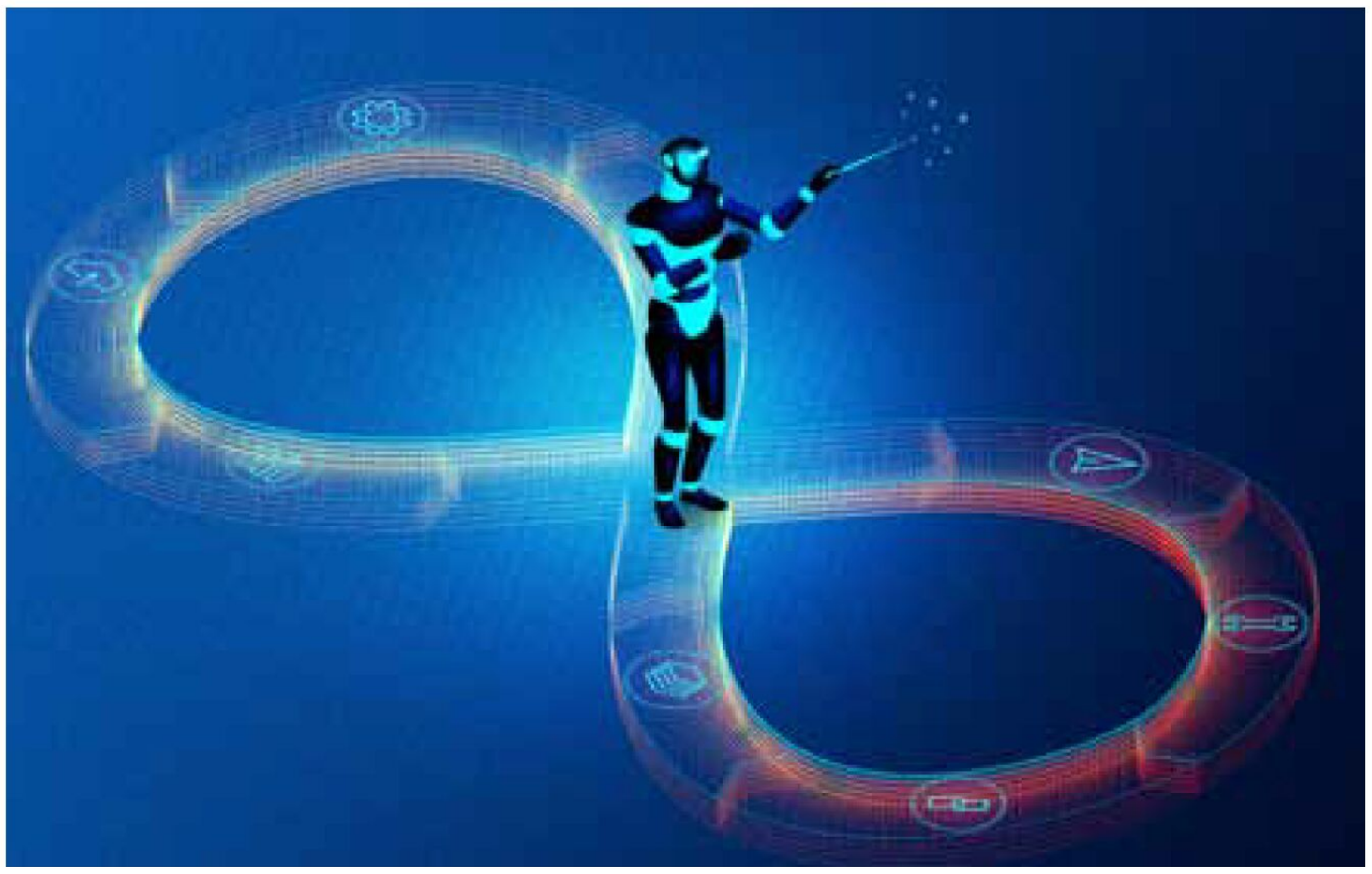
In determining the better approach between DevOps and traditional IT for a stockbroker or healthcare organisation, several factors come into play.

- ***Context matters: Organisation's needs and objectives***
  The choice between DevOps and traditional IT depends on the specific context, goals, and constraints of the organisation. Each approach offers distinct benefits that align with different business requirements.
- ***When DevOps is better***
  DevOps is better suited for a stock broker organisation, as speed and agility are crucial to respond quickly to market fluctuations. A customer-centric focus is needed to deliver personalised trading experiences. Flexibility is essential to adapt to changing market conditions and regulatory requirements.

Continued to page...55

How AI/ML can Automate DevOps

AI/ML integration in DevOps can be transformative for software development. It can enable faster code development, improve the efficiency and reliability of the software, and automate the analysis of massive amounts of data.



The concept of DevOps was born with the purpose of fostering collaboration between the development and operations teams in order to streamline the software development life cycle. Numerous organisations are already reaping the benefits of DevOps. And now, AI and ML are being increasingly used in DevOps pipelines. The aim is to increase the operational efficiency and productivity. A shift towards automation can improve the overall performance of the system. In order to get optimum results, engineering teams adopt AI and ML techniques that can make maximum use of the underlying data and improve the overall DevOps experience.

Steps to include AI/ML in DevOps pipelines

By integrating AI/ML in DevOps, massive amounts of data generated in the DevOps pipeline can be analysed. Different AI/ML algorithms that suit the needs of the organisation can be applied. Here are the key steps:

- Define the goal to use AI/ML in DevOps
- Identify the relevant tools and technologies
- Leverage cloud providers' APIs if available
- Analyse DevOps data and gather appropriate training and testing data
- Correlate different sets of inputs to find the best course of action
- Build and finalise the AI/ML models
- Train the ML models with the DevOps data to match with actuals
- Test model with the test data
- Optimise the model to cater to the dynamic needs of the operations

Areas of application of AI/ML in DevOps

Some notable areas of application of AI/ML in DevOps are:

- Faster code development
  - Tools like GitHub Copilot and AWS Code Whisperer can help expedite code development
  - Identification of irregular code patterns
- Integrated unit test development
  - Creation of automated unit test cases using AI/ML tools
  - Tools like Pynguin can be used to generate unit test cases for Python
- Code debugging
  Automatic detection of bugs in code
- Code reviewing
  - Automated code reviews to identify potential security vulnerabilities, memory leaks, etc
- Robotic process automation
  - Creation of codeless automation macros for regression testing
- Continuous monitoring
  - Identification of behavioural patterns and irregularities
  - Differentiation between normal and abnormal patterns
  - Identification and prioritisation of alerts
  - Identification of hidden patterns and trends
- Anomaly detection
  - Detection of outliers
  - Normalisation of unexpected alerts
  - Detection of DDoS attacks
- Predictive analytics
  - Early actions based on predictive insights

- Capacity planning
  - Efficient capacity planning based on workloads and dynamic system changes
- Optimising resource allocation
  - Overrunning tasks
  - Long build times
- Optimising workflows
  - Creation of workflows aligned to the needs of the business

Challenges of implementing AI/ML in DevOps

Leveraging AI//ML in DevOps should be a planned and gradual approach to avoid common pitfalls. Some of the most common challenges include:

- An AI/ML model trained with improper data can lead to inaccurate results and predictions undermining overall system efficiency.
- DevOps pipelines are hugely customised and there are a plethora of tools available in the market. The AI/ML model that is suitable for some tools may not work well when the organisation migrates to a different set of tools and technologies.
- AI/ML models may need continuous updates to keep with the agility of the processes, leading to frequent updates and added efforts.

The integration of AI/ML in DevOps is one of the key elements of digital transformation. This can rapidly enhance the capabilities of the DevOps processes and bring perceivable value to business. However, AI/ML adoption in DevOps is still evolving and possibilities of failure because of lack of proper implementation still exist. A good business vision, commitment from leadership, and the right change management strategy are essential for the successful implementation of AI/ML in DevOps. END

By: Phani Kiran

The author has nearly 20 years of experience in the IT industry and has worked on digital transformation initiatives.

Continuted to page...53

- ***When traditional IT is better***
  Traditional IT is more suitable for a healthcare organisation as stability and reliability are paramount to ensure patient safety and regulatory compliance. A formal change management process is required to mitigate risks in a sensitive healthcare environment. A risk-averse approach is preferred to prioritise patient well-being.

Considerations for choosing an approach

Several factors influence the decision.

- *Organisational size and complexity:* Larger organisations may benefit from DevOps to streamline operations, while smaller ones may find traditional IT sufficient.
- *Resource availability:* Implementing DevOps may require significant investments in tools and training.
- *Industry and compliance requirements:* Traditional IT aligns with industries with strict regulatory demands, such as healthcare.
- *Organisational culture:* The willingness to embrace change and collaboration in adopting DevOps is a consideration.
- *Hybrid approach:* In some cases, a hybrid approach that combines elements of both DevOps and traditional IT can offer a balanced solution. Organisations can leverage the strengths of each approach to suit their specific needs.

A stockbroker organisation may adopt a hybrid approach by applying DevOps principles for rapid trading platform updates and customer-centric features, while maintaining traditional IT practices for stability in the core infrastructure.

A healthcare organisation could adopt a hybrid approach by using DevOps for non-critical applications, enabling rapid deployment of patient engagement tools, while maintaining traditional IT for critical healthcare systems, ensuring stability and compliance.

The comparison between DevOps and traditional IT in the context of a stockbroker and a healthcare organisation highlights the importance of tailoring IT management approaches to specific industry needs and objectives. DevOps offers speed, collaboration, and customer focus, making it an attractive option for organisations seeking innovation and adaptability. Traditional IT prioritises stability and reliability, making it more suitable for industries with stringent compliance requirements and risk-averse environments.

Ultimately, successful IT management for any organisation requires thoughtful consideration of its unique context and a careful balance between the strengths of DevOps and a traditional IT-like hybrid approach. By understanding the benefits and limitations of each approach, organisations can make informed decisions that optimise software development, IT operations, and ultimately, their ability to deliver value to their customers and stakeholders. END

By: Deepen Dhulla

The author has over 21 years of experience in the Linux and open source realm. He specialises in assisting clients in setting up private clouds using HCI based on ProxmoxVE and Ceph SDS.

DevOps as a Career: Education, Experience and Opportunities

If you are wondering how to go about building a career in the field of DevOps, and what are the skills you need for doing so, read on...



In today's rapidly evolving tech landscape, where software development drives progress, the concept of DevOps has emerged as a pivotal approach that bridges the gap between the development and the operations teams. DevOps, a combination of 'development' and 'operations', is not just a methodology—it's a cultural shift that prioritises collaboration, communication, and automation. As organisations seek to deliver software faster, more reliably, and with higher quality, the demand for skilled DevOps professionals has soared. In this article, we will delve into the world of DevOps as a career opportunity, exploring the motivations, education paths, roles, responsibilities, and skills that define a successful DevOps engineer.

Why pursue a career in DevOps

Bridging the gap between development and operations

In traditional software development environments, development and operations teams often operated in isolation, leading to communication gaps, longer development cycles, and challenges in ensuring software quality. DevOps aims to break down these silos by fostering collaboration and shared responsibilities between the two teams. By aligning goals and practices, DevOps enables a streamlined development process, leading to faster innovation and reduced time-to-market.

Enhancing software development life cycle

One of the primary reasons for the widespread adoption of DevOps is its significant impact on the software development life cycle. DevOps practices, such as continuous integration and continuous deployment (CI/CD), automate the process of building, testing, and deploying software. This automation not only reduces manual errors but also accelerates the development cycle, allowing organisations to quickly respond to market demands and user feedback.

High demand and competitive salaries

The rise of DevOps has led to a surge in demand for skilled professionals who can effectively implement its principles. According to the DevOps Institute's Upskilling Report, 70% of organisations surveyed are actively seeking DevOps professionals. This demand has also translated into attractive salaries. DevOps engineers often command competitive remuneration due to their role in driving organisational efficiency and software excellence. According to industry reports, the average salary for a DevOps engineer ranges from $80,000 to $130,000 per year, with significant variations based on experience and location.

Insight: Industry growth and demand

According to the DevOps Institute's Upskilling Report, over 70% of organisations surveyed reported a significant demand for DevOps professionals. This demonstrates the increasing recognition of DevOps as an essential practice in modern software development.

How to start a career in DevOps

Acquiring relevant education and training

Starting a career in DevOps requires a strong foundation in both development and operations practices. While formal education is beneficial, practical knowledge gained through hands-on experience is equally crucial. Many universities and online platforms offer degrees and certifications in DevOps-related fields. Industry-recognised certifications like the 'Certified DevOps Engineer' from Amazon Web Services (AWS) and 'DevOps Engineer' from Google Cloud can enhance your credibility in the job market.

Gaining hands-on experience

Academic knowledge is best complemented by hands-on experience. Engaging in internships, contributing to open source projects, or working on personal projects helps you apply theoretical concepts in real-world scenarios. Building your portfolio with practical examples demonstrates your ability to navigate DevOps tools and practices effectively.

Insight: Job market trends

LinkedIn's 'Emerging Jobs Report' revealed that 'DevOps Engineer' was among the top emerging job roles, experiencing rapid growth year after year. This data underlines the industry's emphasis on DevOps expertise.

Mastering key DevOps tools and technologies

DevOps thrives on automation and modern tools. Familiarity with tools like Jenkins for continuous integration, Docker for containerisation, and Kubernetes for orchestration is essential. Cloud platforms like AWS, Azure, and Google Cloud are integral to modern DevOps practices. Proficiency in scripting languages like Python and shell scripting is also valuable for automation and configuration management.

Where to find DevOps opportunities

Tech companies and startups

Large tech companies and startups alike are embracing DevOps to streamline their development processes and improve software delivery. DevOps engineers in these organisations play a pivotal role in orchestrating complex development pipelines, ensuring scalability, and optimising infrastructure.

Cloud service providers

Cloud service providers like AWS, Azure, and Google Cloud have become essential enablers of DevOps practices. DevOps engineers leverage cloud services to build scalable and reliable infrastructure. For instance, AWS's 'CloudFormation' enables Infrastructure-as-Code (IaC), allowing DevOps engineers to define and provision resources programmatically.

Consulting firms and DevOps service providers

Consulting firms specialise in helping organisations adopt and implement DevOps practices. DevOps engineers in consulting firms work on a variety of projects across different industries, helping clients transform their development and operations processes. These engineers collaborate with teams to design and implement efficient CI/CD pipelines and establish best practices.

Insight: Impact on software quality

A study conducted by Puppet and DORA (DevOps Research and Assessment) demonstrated that high-performing DevOps organisations deploy 46 times more frequently and have a change failure rate that is 7.5 times lower. This indicates the direct correlation between DevOps practices and enhanced software quality.

Roles, responsibilities, and skills of a DevOps engineer

Understanding the role of a DevOps engineer

DevOps engineers plays a crucial role in aligning development and operations activities. They are responsible for automating manual tasks, optimising deployment pipelines, and ensuring that software releases are efficient and reliable. DevOps engineers champion the use of CI/CD pipelines, enabling developers to commit code changes that are tested, integrated, and deployed automatically.

Collaboration and communication

DevOps engineers serve as bridges between different teams. Effective communication and collaboration skills are vital, as they interact with developers, operations teams, and other stakeholders. They ensure that the entire development life cycle is well-coordinated, from writing code to deploying applications to production environments.

Automation and Infrastructure as Code (IaC)

Automation lies at the core of DevOps practices. DevOps engineers design and implement automated workflows that streamline repetitive tasks, enabling faster software delivery. Infrastructure as Code (IaC) is a key concept that involves managing infrastructure using code scripts. DevOps engineers use tools like Terraform and Ansible to automate the provisioning and management of infrastructure resources.

Use cases and examples

Case studies in successful DevOps implementations

Numerous success stories demonstrate the transformative power of DevOps. Take the case of Etsy, an e-commerce platform. By adopting DevOps practices, Etsy reduced their deployment time from hours to minutes and achieved higher availability. Another example is Netflix, which credits its ability to release multiple updates daily to its robust DevOps culture.

Use cases of CI/CD in action

Continuous integration and continuous deployment (CI/CD) pipelines are fundamental to DevOps. Consider an online retail platform. A DevOps engineer designs a CI/CD pipeline that automatically tests and deploys code changes. This ensures that new features and fixes are quickly delivered to users, enhancing user experience and business agility.

Analysis: DevOps skills and salary

The '2021 DevOps Salary Report' by DevOps Institute showed that professionals with DevOps skills earn higher salaries. For instance, those with the 'Certified DevOps Engineer' certification earned an average of 20% more than their non-certified counterparts.

Case study: Netflix

Netflix's DevOps culture is renowned. Their ability to release updates multiple times a day has transformed the company's approach to software delivery. By automating processes and fostering a collaborative culture, Netflix remains at the forefront of DevOps excellence.

The road ahead

The journey to becoming a proficient DevOps engineer is marked by dedication and passion. DevOps engineers are the architects of the future, shaping how software is conceived, constructed, and maintained. As industries continue to evolve in response to technological advances, the demand for skilled DevOps professionals will only intensify. This demand isn't just a reflection of technical prowess; it's a testament to the ability to drive transformational change and foster innovation.

Skills and mindset

DevOps isn't merely a job; it's a mindset. Beyond technical skills, successful DevOps engineers possess a holistic view of the software development life cycle. They understand the importance of continuous improvement and embrace challenges as opportunities for growth. A problem-solving attitude and adaptability to rapidly changing technologies are crucial in this dynamic field.

Continuous learning and evolution

DevOps engineers must remain perpetual learners. The tools, platforms, and methodologies that define DevOps today might evolve tomorrow. Staying current with industry trends, attending conferences, and engaging with the DevOps community are all vital components of a successful DevOps career.

DevOps stands as a beacon of innovation, efficiency, and collaboration. The convergence of development and operations has reshaped how software is conceptualised, created, and delivered. DevOps engineers play a vital role in this landscape, orchestrating the delicate symphony between code creation and deployment, automating processes that were once manual, and fostering a culture of collaboration. END

By: Meghraj Singh Beniwal

The author is the founder of Schooling for Next Generation, a tech enthusiast and Android hobbyist. He is a test automation analyst working in Virginia in the US.

R Series: mlr3 Package

In this 24th post in our ongoing exploration of the R series, we dive into the mlr3 package.

The mlr3 package offers an object-oriented programming implementation for machine learning in R. It is a successor to the mlr package, and has only the essential building blocks for machine learning. It moves away from R's S3 classes and uses the R6 encapsulated classes with reference semantics. The software contains only computational code and algorithms, and leaves the visualisation to extra packages. The return types are documented, and type safety is provided. It uses *data.table* for representing tabular data and is highly efficient and performant. The number of package dependencies is greatly reduced, and hence installation and maintenance are easier compared to the mlr package. We will use R version 4.2.2 installed on Parabola GNU/Linux-libre (x86-64) for the code snippets.

```
$ R --version

R version 4.2.2 (2022-10-31) -- "Innocent and Trusting"
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/.
```

To begin, you can install and load the 'mlr3' package using these commands:

```
> install.packages("mlr3")

Installing package into '/home/shakthi/R/x86_64-pc-linux-gnu-
library/4.1'
(as 'lib' is unspecified)
...
 *** installing help indices
 *** copying figures
 ** building package indices
 ** testing if installed package can be loaded from temporary
location
 ** testing if installed package can be loaded from final
location
 ** testing if installed package keeps a record of temporary
installation path
 * DONE (mlr3)

> library("mlr3")
```

mlr_tasks

The tabular data and metadata information for a machine learning problem are stored in objects called 'tasks' in mlr3. There are 11 data sets available in this version of the mlr3 package as shown here:

```
> mlr_tasks
<DictionaryTask> with 11 stored values
Keys: boston_housing, breast_cancer, german_credit, iris,
mtcars,
  penguins, pima, sonar, spam, wine, zoo
```

You can obtain a task from the dictionary using the *tsk()* function. An example task for the mtcars data set is given below:

```
> task_mtcars = tsk("mtcars")
```

```
> task_mtcars
<TaskRegr:mtcars> (32 x 11): Motor Trends
 * Target: mpg
 * Properties: -
 * Features (10):
  - dbl (10): am, carb, cyl, disp, drat, gear, hp, qsec, vs,
wt
```

The boston_housing data set contains information on various houses in Boston such as the proportion of non-retail business acres per town (indus), the percentage of lower status of the population (lstat), index of accessibility to radial highways (rad), and the average number of rooms per dwelling (rm), etc.

```
> task_boston_housing = tsk("boston_housing")

> task_boston_housing
<TaskRegr:boston_housing> (506 x 19): Boston Housing Prices
 * Target: medv
 * Properties: -
 * Features (18):
  - dbl (13): age, b, cmedv, crim, dis, indus, lat, lon,
lstat, nox,
    ptratio, rm, zn
  - int (3): rad, tax, tract
  - fct (2): chas, town
```

The *data.table* function

The *data.table()* function is implemented in the data.table package for storing tabular data. It is an alternative to R's data.frame, and provides more functionality and better performance. For example:

```
> dt = as.data.table(task_boston_housing)

> dt
     medv  age      b chas cmedv    crim    dis indus     lat
lon lstat
  1: 24.0 65.2 396.90    0  24.0 0.00632 4.0900  2.31 42.2550
-70.9550  4.98
  2: 21.6 78.9 396.90    0  21.6 0.02731 4.9671  7.07 42.2875
-70.9500  9.14
  3: 34.7 61.1 392.83    0  34.7 0.02729 4.9671  7.07 42.2830
-70.9360  4.03
  4: 33.4 45.8 394.63    0  33.4 0.03237 6.0622  2.18 42.2930
-70.9280  2.94
  5: 36.2 54.2 396.90    0  36.2 0.06905 6.0622  2.18 42.2980
-70.9220  5.33
...
```

The *as_task_regr* function

A regression task needs to be created using the *as_task_regr()* function for a given data.frame type object. The *subset()* function is used to retrieve selective data from the data set. In this example, the 'cylinder' and 'gear' columns from the mtcars data set are selected, and the gear column is chosen as the prediction field:

```
> mtcars_set = subset(mtcars, select = c("cyl", "gear"))

> str(mtcars_set)
'data.frame':  32 obs. of  2 variables:
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...

> task_mtcars = as_task_regr(mtcars_set, target = "gear", id
= "car")

> task_mtcars
<TaskRegr:car> (32 x 2)
 * Target: gear
 * Properties: -
 * Features (1):
  - dbl (1): cyl
```

The *head* function

The first few entries of the data set can be viewed using the *head()* function as illustrated:

```
> task_mtcars$head()
   gear cyl
1:    4   6
2:    4   6
3:    4   4
4:    3   6
5:    3   8
6:    3   6
```

The *summary* function

The *summary()* function can also be used on the data set to display the quartiles, minimum, maximum, median and mean values as illustrated here:

```
> summary(as.data.table(task_mtcars))
      gear            cyl
 Min.   :3.000   Min.   :4.000
 1st Qu.:3.000   1st Qu.:4.000
 Median :4.000   Median :6.000
 Mean   :3.688   Mean   :6.188
 3rd Qu.:4.000   3rd Qu.:8.000
 Max.   :5.000   Max.   :8.000
```

The *data* function

The complete data for the regression task can be viewed using the *data()* function as shown here:

```
> task_mtcars$data()
    gear cyl
 1:    4   6
 2:    4   6
 3:    4   4
 4:    3   6
 5:    3   8
 6:    3   6
 7:    3   8
 8:    4   4
 9:    4   4
...
```

You can also limit the number of rows and columns to be shown as illustrated below:

```
> task_mtcars$data(rows = c(1, 2, 3), cols = task_
mtcars$feature_names)
   cyl
1:   6
2:   6
3:   4
```

Data display with *nrow, ncol*

The data dimensions can be displayed using *nrow* and *ncol* as shown in this snippet:

```
> c(task_mtcars$nrow, task_mtcars$ncol)
[1] 32  2
```

The features and the target column names are fetched using *feature_names* and *target_names*, respectively, as illustrated here:

```
> c(Features = task_mtcars$feature_names, Target = task_
mtcars$target_names)
Features   Target
   "cyl"   "gear"
```

The *select* and *filter* functions

The *select()* function can limit the desired features to be used. In the following example, the 'cylinder' column is alone chosen from the mtcars regression.

```
> task_mtcars$select("cyl")

> task_mtcars
<TaskRegr:car> (32 x 2)
 * Target: gear
 * Properties: -
 * Features (1):
  - dbl (1): cyl
```

The rows in the data set can be selected using the *filter()* function as demonstrated below:

```
> task_mtcars$filter(1:3)

> task_mtcars$data()
   gear cyl
1:    4   6
2:    4   6
3:    4   4
```

The *partition* function

The *partition()* function is used to split the data set into training and test data. For the given mtcars data set, we get 21 entries for the training set, and 11 data points for testing.

```
> subset = partition(task_mtcars)

> subset
$train
 [1]  3  5  8  9 10 21 27 30 32  7 12 14 16 22 23 24 29 31
19 26 28

$test
 [1]  1  2  4 25  6 11 13 15 17 18 20
```

Training data

The Learner class provides many machine learning algorithms in R. In this example, we will use the *lrn()* regression tree learner for the mtcars data set and training data:

```
> learning = lrn("regr.rpart")
> learning$train(task_mtcars, row_ids = subset$train)

> learning$model
n= 21

node), split, n, deviance, yval
      * denotes terminal node

1) root 21 637.86950 19.79524
  2) cyl>=5 12  74.58250 15.72500 *
  3) cyl< 5 9  99.41556 25.22222 *
```

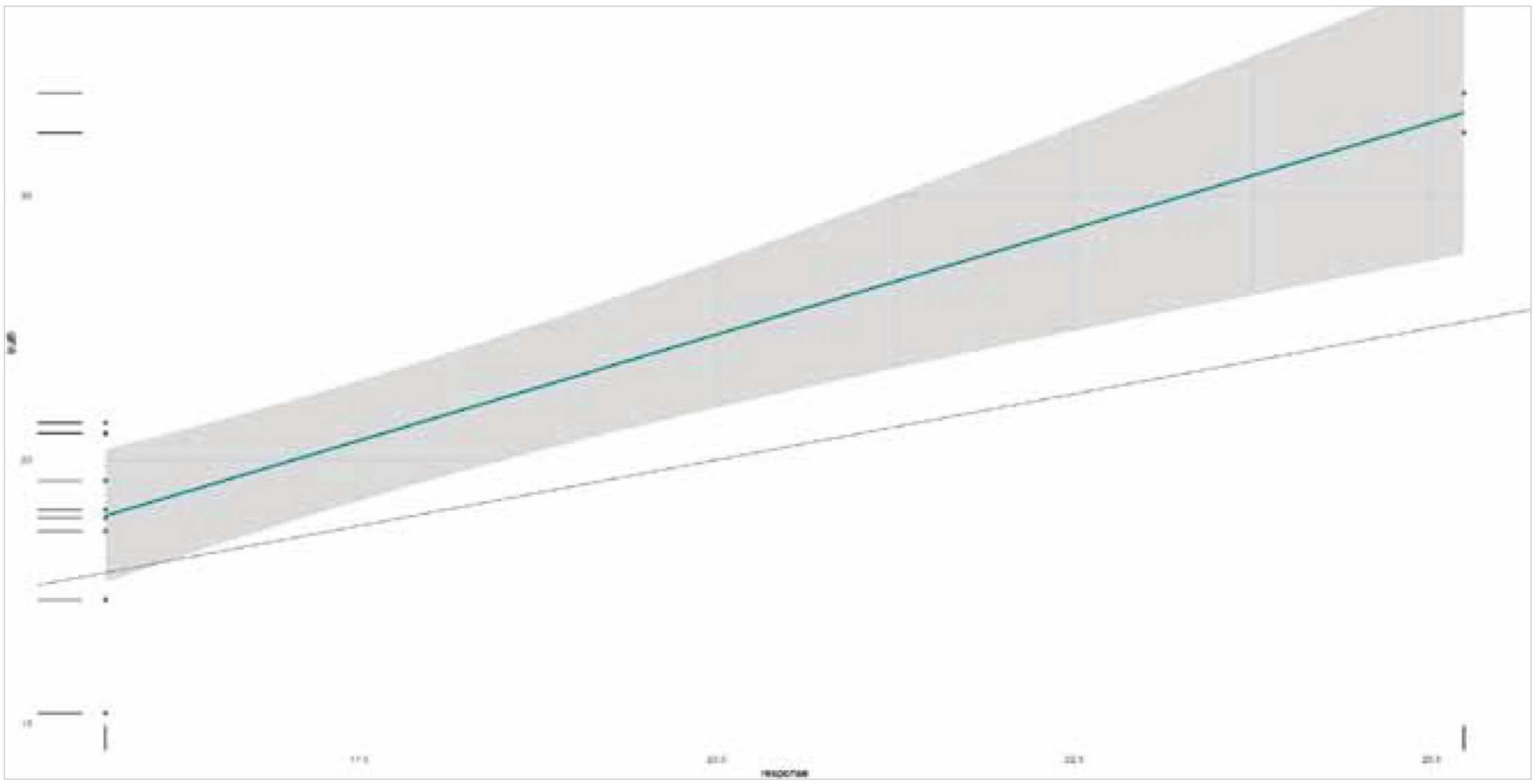


Figure 1: Autoplot

The *predict* method

The *predict()* method can be used for prediction from the training model for the test data as shown below:

```
> predict = learning$predict(task_mtcars, row_ids =
subset$test)

> predict
<PredictionRegr> for 11 observations:
    row_ids truth response
          1  21.0 15.72500
          2  21.0 15.72500
          4  21.4 15.72500
---
         17  14.7 15.72500
         18  32.4 25.22222
         20  33.9 25.22222
```

The *autoplot* method

The *autoplot()* method can be used to visualise the prediction object. However, you need to first install the mlr3verse package as shown here:

```
> install.packages("mlr3verse")
Installing package into '/home/shakthi/R/x86_64-pc-linux-gnu-
library/4.1'
...
 ** building package indices
 ** testing if installed package can be loaded from temporary
location
 ** testing if installed package can be loaded from final
location
 ** testing if installed package keeps a record of temporary
installation path
 * DONE (mlr3verse)
```

You can then load the 'mlr3viz' library, and run the *autoplot* function on the prediction result. Figure 1 shows the predicted and truth values for the mtcars data set.

```
> library(mlr3viz)

> predict = learning$predict(task_mtcars, row_ids =
subset$test)

> autoplot(predict)
```

You are encouraged to read the mlr3 package reference manual to learn more about functions, arguments and usage. END

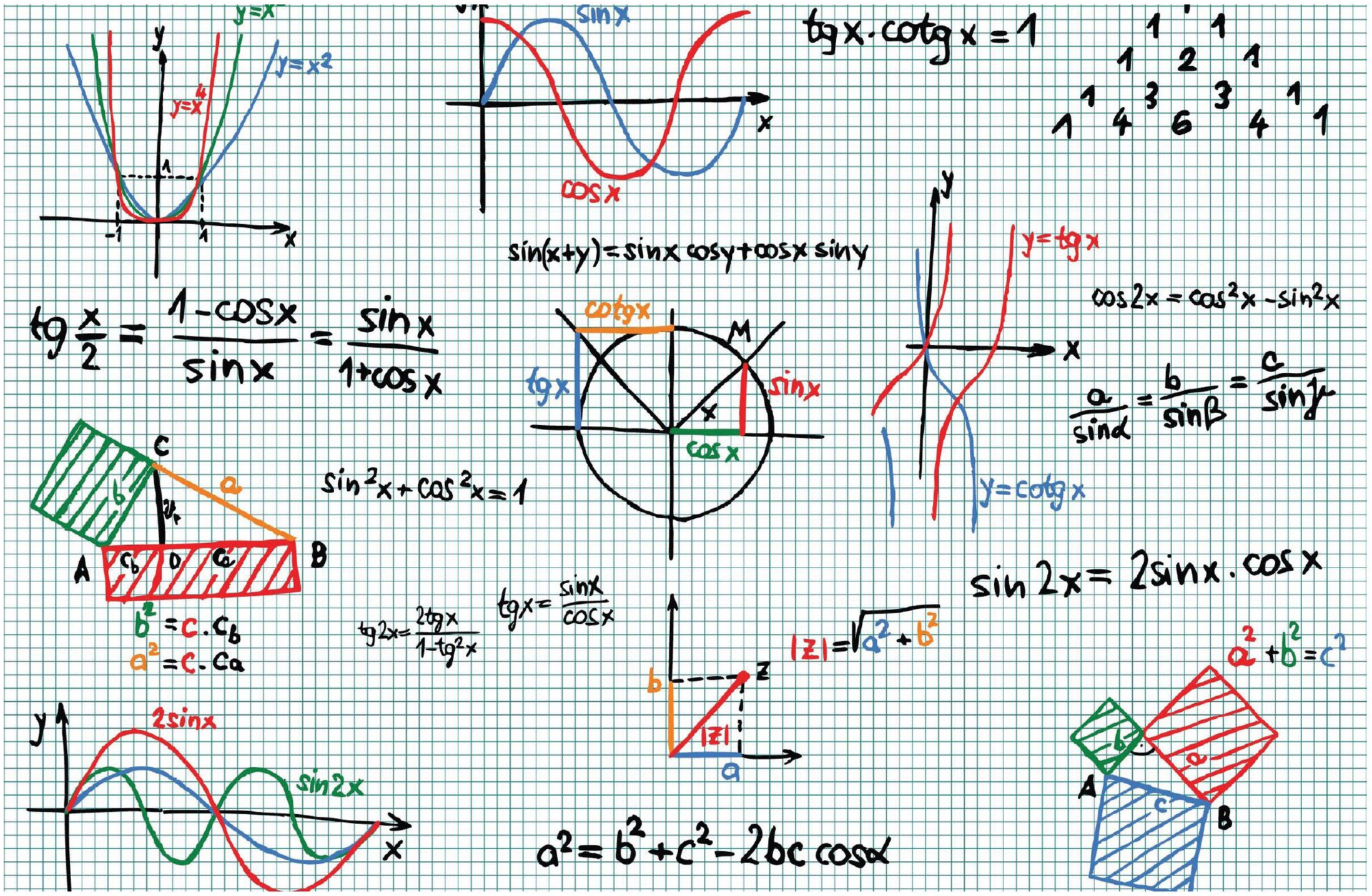
References

- mlr3: Machine Learning in R; *https://mlr-org.com/*
- mlr3 reference; *https://mlr3.mlr-org.com/reference/*

By: Shakthi Kannan

The author is a free software enthusiast.

Coq and Lean: Powerful Interactive Theorem Provers



Coq and Lean are tools that elevate trust in research and software development with rigorous proofs. They also help to prove that code is bug-free.

It is undeniable that the open source software movement is affecting the world positively. But how exactly does open source software impact human lives? First and foremost, it offers popular software for free (free as in 'free beer'). Nowadays, you don't have to pay for an operating system or a word processing package. There are tons of open source software freely available for you to choose from. This shift has also prompted software giants to offer their software at relatively lower prices to remain competitive.

However, the impact of open source software extends far beyond this. For instance, the research community in various fields, including science, engineering, technology, management, and social sciences, has greatly benefited from the open source software movement. Notable examples include software like Scilab, SageMath, Project Jupyter, LaTeX, and Python libraries like NumPy and SciPy, among others. The list is so extensive that attempting to enumerate all of them would be futile. In fact, I think I have left out many important names from the list. What makes this even more fascinating is the fact that the open source community boasts some of the world's most brilliant minds contributing as programmers and software engineers—talent that even the largest software companies can only dream of employing. As a result, highly specialised and

technical software has been developed to cater to the needs of communities engaged in top-class research in science, engineering, and technology.

In this article, we will acquaint ourselves with one such category of software: interactive theorem provers.

Undoubtedly, all of us have had to prove some theorems in our lifetime, with the Pythagorean theorem being a prime example. But why do we need interactive theorem provers? Imagine that you are a researcher confident of proving an essential theorem. However, how can the research community fully trust your proof? There might be an unnoticed error in your reasoning. This is where peer review comes to our rescue. To gain credibility, you must submit your proof to a reputable journal, where experts in your specific area of research will evaluate your work and verify its accuracy and authenticity. However, human error can still creep in, and indeed, such issues have occurred in the research community on numerous occasions.

The history of the four-colour theorem

Let us briefly go through the history of the famous four-colour theorem as an example of erroneous proofs being mistakenly accepted as correct. The four-colour theorem states that no more than four colours are required to colour the regions of any map so that no two adjacent regions sharing a boundary will have the same colour. Notice that if two or more regions meet only at a point, then they can share the same colour. Figure 1 shows an example of a map coloured in just four colours (*Courtesy:* Wikipedia). The observation was first made by Francis Guthrie in 1852. In 1879, a proof was proposed by Kempe. This proof was widely accepted as correct. However, in 1890, Heawood proved that Kempe's proof was incorrect. Notice that an incorrect proof was believed to be true for almost 11 years.

The work by Kempe was used by Heawood to prove a weaker theorem known as the five-colour theorem. After many false and incomplete proofs and almost a century later, the four-colour theorem was finally proved in 1976 by Appel and Haken. However, their proof was computer-based, and due to the immense size of this proof, its veracity was still doubted by many. A simpler proof using the same ideas (again relying on computers) was published in 1997 by Robertson, Sanders, Seymour, and Thomas. Finally, in 2005, the theorem was proved by Werner and Gonthier using Coq, thus ending any doubts about the proof of the theorem. Now, we can place our trust solely in the Coq kernel. So, in a nutshell, an interactive theorem prover can be a powerful tool in establishing the validity of complex theorems.

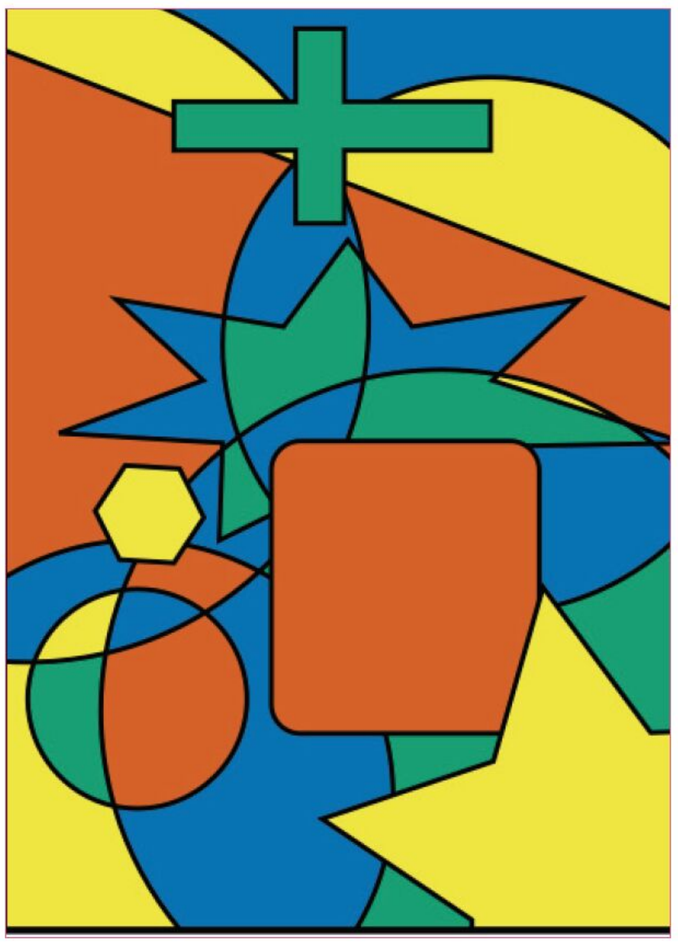


Figure 1: Example of a four-coloured map (*Courtesy:* Wikipedia)

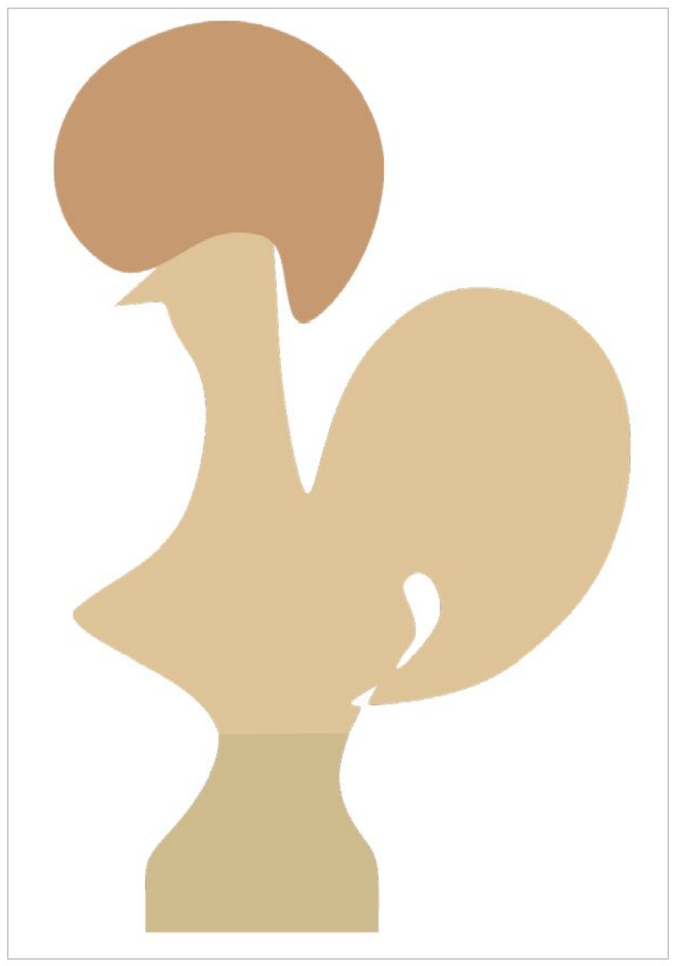


Figure 2: Logo of Coq

In this article, we will only introduce two interactive theorem provers: Coq and Lean. Although the article will not delve into the details of these two software, it will serve an essential purpose. The readers will become aware of the abundance of solutions offered by the open source software community, and they will also realise that solutions may exist even for the most difficult problems.

Coq

Coq is free and open source software developed by INRIA, a French research institute focusing on computer science and applied mathematics. Coq operates under the LGPLv2.1 licence. It was first released in 1989, and the latest stable version is version 8.17.1. Figure 2 shows the logo of Coq. As mentioned earlier, a definitive proof of the four-colour theorem is one of the early success stories of Coq. The Feit–Thompson theorem is another important theorem proved using Coq. Coq was developed using Ocaml, which is a multi-paradigm programming language with emphasis on functional programming.

Now, let us see how to use Coq from our system. One good thing is that you don't need to install Coq on your system to work with it. You can run it in your browser using a web-based environment called jsCoq. This environment is available online at *https://coq.vercel.app*. Figure 3 shows the terminal of the jsCoq environment. For beginners, I suggest using this web-based environment.

However, Coq is also cross-platform software; it can be installed and used on Windows, macOS, and Linux. In Linux, Coq can be used with different interfaces like Vim, Emacs, etc. As an example, we will see how to install and use Coq in Ubuntu. I have used Proof General, an Emacs interface for Coq. Usually, Emacs is installed in Ubuntu by default;

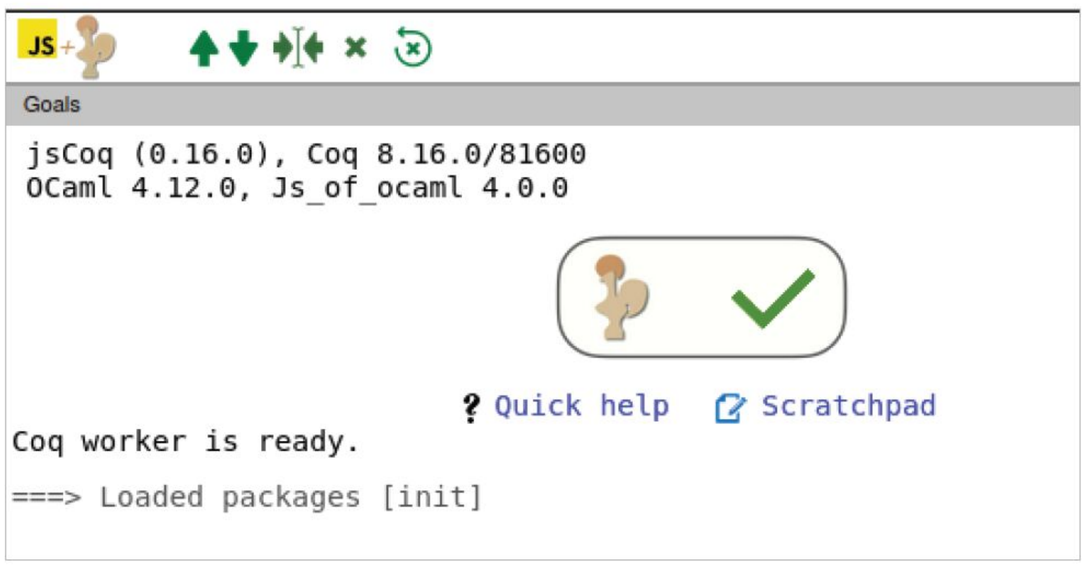


Figure 3: The terminal of the Coq online environment

otherwise, you need to install Emacs on your system. Emacs, Proof General, and Coq can be installed on your system by executing the following three commands in the terminal:

```
sudo apt install emacs
sudo apt install proofgeneral
sudo apt install coq
```

Now, open the Emacs terminal by executing the command 'emacs' in the terminal. From this point onwards, we have to use some basic Emacs commands to use Proof General and Coq. This is the reason why I recommended the web-based jsCoq over Proof General for working with Coq. In this way, some of the complexities associated with Emacs can be avoided while handling the intricacies of Coq. However, notice that the Coq syntax used is the same in both these environments.

Once the Emacs terminal is opened, press 'Ctrl+x' and then 'Ctrl+f' to open a file in Emacs. When prompted, give a name to the Coq file in which your Coq source code should be saved. In this example, I have used the file name 'coq1.v'. Notice that Coq files have a '.v' extension. Now, you will see Proof General coming into action. Figure 4 shows the window of Proof General displayed when a Coq file is opened.

Later, a file named 'coq1.v' will be opened in the Emacs terminal. In the file, enter the following line of code: *'Print nat.'* to print the definition of 'nat'. Keep in mind that Coq is case-sensitive, and the dot (.) after the code *'nat'* is mandatory. Now, on the Emacs terminal, press 'Ctrl+c' and then 'Ctrl+n' to run the next step in the Coq program. Since there is just one line in our Coq program, that line will be executed, and the output will be displayed. Figure 5 shows the Emacs terminal when this simple Coq program is executed. The same line of code can also be executed in the web-based environment jsCoq.

The following Emacs commands are also useful for processing Coq programs. To go to the previous step in a Coq program, press 'Ctrl+c' and then 'Ctrl+u'. Press 'Ctrl+x' and then 'Ctrl+s' to save the changes made to the current Coq program being edited. Finally, press 'Ctrl+x' and then 'Ctrl+c' to close the Emacs terminal. After executing this command, type 'yes' when prompted to quit.



Figure 4: Proof General window of Coq

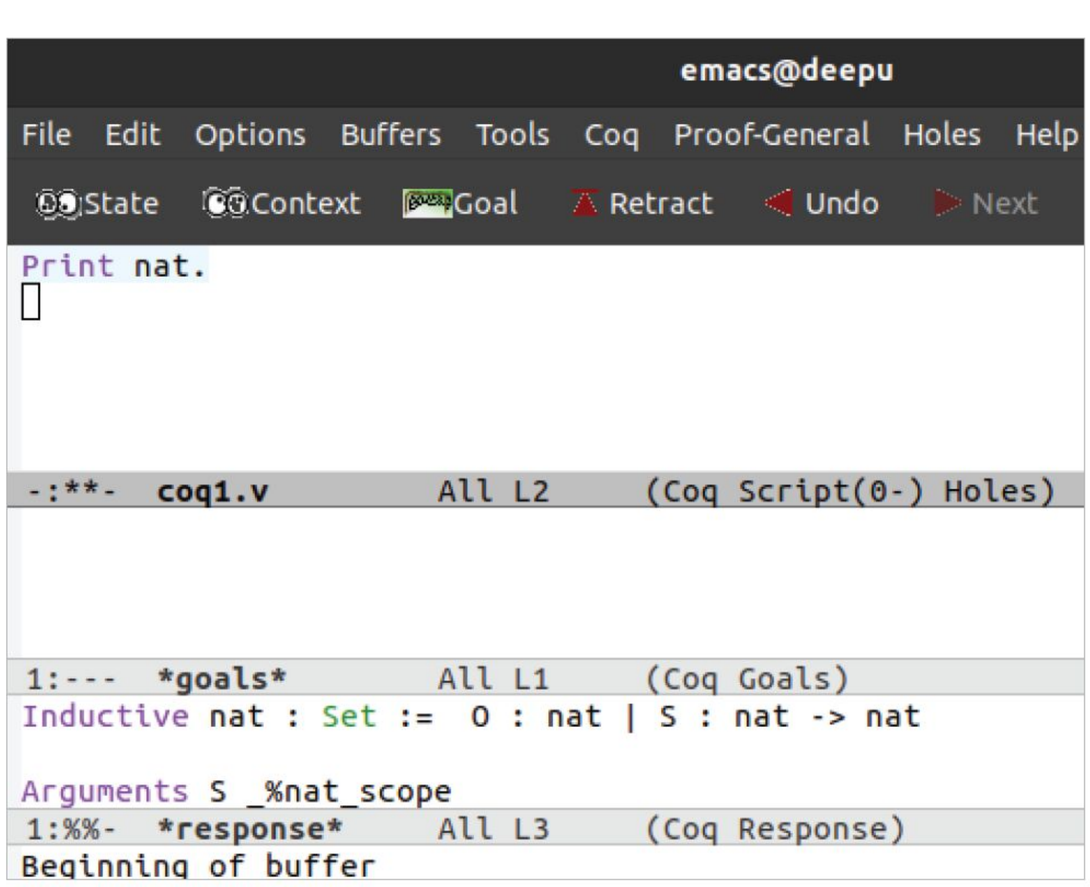


Figure 5: A simple Coq program

But what is the meaning of the output shown in Figure 5, which is obtained when the single line of Coq code got executed? The line of code on execution prints the definition of natural numbers (the set of numbers 0, 1, 2, …, denoted by the symbol N). Notice that if you are using jsCoq, when you type 'nat' on the environment, it is automatically replaced with the symbol N. From the definition printed on the terminal, we can see that 0 is defined as a natural number and the successor

of a natural number is also defined as a natural number. Thus 1 is a natural number because 0 is defined as a natural number, and so on and so forth. This definition is based on the Peano axioms for natural numbers. To keep things simple, I will refrain from introducing any technical terms in the rest of our discussion. Further, it is beyond the scope of this article to discuss even simple Coq programs proving very basic theorems like De Morgan's laws. Therefore, we conclude our discussion of Coq by highlighting its main components and the strategy used in giving proofs using Coq.

Coq involves two types of commands: 'vernacular' and 'tactics'. Vernacular commands allow interaction with the Coq environment and include commands like *Theorem*, *Lemma*, *Fact*, etc. Tactics, on the other hand, are used to interact with the proof state and guide the proof process. They apply various proof strategies, such as simplifying expressions, applying logical rules, performing case analysis, and introducing hypotheses. Examples of tactical commands include *intros*, *rewrite*, and *destruct*. Coq also provides a specification language called Gallina. Interestingly, the word 'Coq' means 'rooster' in French, while 'Gallina' means 'hen' in Latin. Additionally, Coq has a built-in tactic scripting language called Ltac, which automates and customises the proof process. While Ltac is widely used with Coq, it is not often considered a standard part of Coq.

Now, let us discuss the strategy used in Coq to prove a theorem. Coq provides constructive proofs. It has a type called 'True', which consists of the set of all provably true propositions. Thus, in Coq, if you can find a value for a type, then it is considered proved. For example, the type 'nat', as seen in the line of code discussed earlier, proves that natural numbers do exist. The proof strategy in Coq follows this approach: a required theorem to be proven is denoted as a type. A proof for this theorem involves deriving a value for this type based on fundamental facts. This proof strategy also ensures that a false value can never be derived in Coq. The reason is that a value of 'False' is impossible to construct in Coq.

Lean

Now, let us turn our attention to Lean. Nowadays, we frequently encounter the names of companies known for developing proprietary software even when discussing free and open source software. Therefore, it comes as no surprise that Lean, the second interactive theorem prover we are discussing in this article, is developed by Microsoft Research. The project, primarily developed by Leonardo de Moura, was first released in 2013. Lean is free and open source software licensed under the Apache License 2.0. Figure 6 shows the logo of Lean.

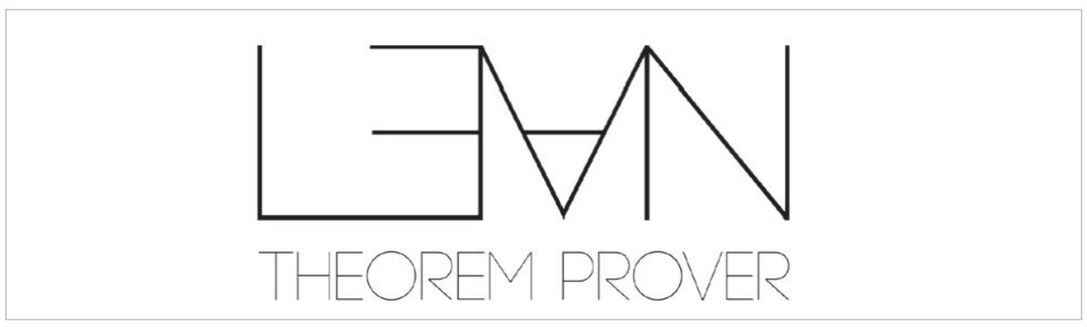


Figure 6: The logo of Lean

At present, there are two versions of Lean available: Lean 3 and Lean 4. However, it's important to note that Lean 4 is not compatible with Lean 3, meaning that code written for one version may not work in the other. One of the standout features of Lean is its powerful mathematical library known as *mathlib*. Initially, Lean 4 encountered issues when utilising this library, but these have now been resolved, and Lean 4 can seamlessly use *mathlib* as well. If you prefer to use Lean 4 without installing it in your system, you have the option of utilising a web-based environment accessible at *https://lean.math.hhu.de*. However, should you decide to install and run Lean 4 on your local machine, the most straightforward approach is to use the Visual Studio Code editor (VS Code editor). Although I won't delve into the detailed steps for setting up Lean 4 in VS Code editor here due to space constraints, I assure you that the installation and usage process is remarkably easy and straightforward.

Now, let us explore how Lean 4 code can be executed in both the VS Code editor and the web-based environment. Take a look at the line of code '*#check 5 + 5*'. Notice that Lean programs are saved with a '.lean' extension. When you run this code, it will produce the output '*5 + 5 : Nat*', confirming that the number 10 is also a natural number.

It is time to wind up this article, but before we conclude our discussion, two important questions need to be addressed. First, are there any notable distinctions between Coq and Lean in their approach to obtaining a proof? I have been informed that Lean is better suited while using classical logic to derive a proof. However, Coq has a larger and older community around it when compared to Lean. Nevertheless, any disparities between the two will diminish over time since both software packages are experiencing rapid development. Second, are there any other applications for interactive theorem provers aside from proving theorems? Absolutely! Both Coq and Lean can be used to provide guarantees for the programs you have developed. Rather than relying on test cases to verify a program's correctness, you can utilise Coq or Lean to formally prove that the program is free of bugs. END

Acknowledgment:

I thank Julin Shaji, a former student of mine, for teaching me the basics of Coq. While preparing this article, I have referred to his notes (a work in progress) on Coq.

By: Dr Deepu Benson

The author is a free software enthusiast and his area of interest is theoretical computer science. He maintains a technical blog.

Using Apache Iceberg for Developing Modern Data Tables

Apache Iceberg, a data management powerhouse, empowers businesses with its remarkable flexibility, seamless tool compatibility, and endorsement by the major tech players in the industry.



The cloud has made it possible to collect vast quantities of data and store it at a reasonable cost. It is also responsible today for the emergence of data lakes, data mesh and other modern data architectures. However, handling large amounts of data using a generic cloud offers its own challenges and limitations.

For instance, query engines often cannot react with typical blob storage systems smoothly. To address this issue, applying a table format to data becomes extremely useful.

Table format data was originally developed by Netflix and was open sourced as an Apache incubator project in 2018. It graduated from this in 2020 and became Apache Iceberg. Apache Iceberg can meet certain challenges faced by Apache Hive.

Process engines used by a company may change from time to time. For example, many business firms have moved from Hadoop to Spark or Trino. Iceberg provides greater flexibility and choice for processing engines.

Additionally, data lakes have undesirable qualities due to table formats obtained from older technologies. Apache Iceberg is built to address such shortcomings in Apache Hive. Moreover, Iceberg also supports multiple file formats, including Apache Parquet, Apache Avro and Apache ORC. Hence it provides more flexibility and long-term adaptability for file formats that may emerge in the future.

Reasons to choose Apache Iceberg

Apache Iceberg is a new open table format developed by the open source community. The modern data industry needs many metadata tables such as data files tables, all manifest tables, all entries tables, history tables, partition tables, and snapshot tables, etc. Iceberg provides an effective solution for the creation of such tables.

Table 1 compares several features of three major data lake table format creators: Apache Iceberg, Apache Hudi and Delta Lake.

Factor	Apache Iceberg	Apache Hudi	Delta Lake
ACID transaction	Present	Present	Present
Partition evolution	Present	Absent	Absent
Schema evolution	Present	Partial	Limited
Time-travel	Present	Present	Present
Project governance	Apache projects with a diverse PMC (top level project)	Apache projects with a diverse PMC (top level project)	Linux foundation projects with all data bricks TSC
Tool read compatibility	Apache Hive, Demio Sonar, Apache Flink, Apache Spark, Presto, Trino, Athena, Snowflake, Databricks Spark, Apache Impact, Apache Drill	Apache Hive, Demio Sonar, Apache Flink, Apache Spark, Presto, Trino, Athena, Databricks Spark, Apache Impact, Red Shift, Big Query	Apache Hive, Demio Sonar, Apache Flink, Databricks SQL, Analytics, Presto, Trino, Athena, Snowflake, Red Shift, Apache Bea
Tool write compatibility	Apache Hive, Demio Sonar, Apache Flink, Apache Spark, Presto, Trino, Athena, Snowflake, Databricks Spark, Debezium	Apache Flink, Apache Spark, Databricks Spark, Debezium, Kafka Connect	OSS Delta Lake, Apache Spark, Trino, Apache Flink, Databricks Spark, Kafka Connect, Debezium, Distributed Delta Lake

Table 1: A comparison of Apache Iceberg, Apache Hudi and Delta Lake (Table credit: *dremio.com*)

One of the important features of Iceberg is partitioning, a technique used for optimising data lakes. For example, for sales data where queries often rely on when the sales occurred, you may partition data by the day, month, or year of the sale.

Data lakes aim to democratise data. However, it is necessary to reduce the complexity of data structures and physical data storage. In 2009, Facebook released a Hive table format that could not answer this problem. In contrast, Iceberg, however, gives table format specifications with a set of APIs and libraries to interact with tables.

Unless Apache Hive adds it, removing and updating data is very fast and easy in Iceberg. It supports SQL commands. So, it can be used to work with Hive, Spark, Impala, and so on. Such table formats are becoming the format of choice for large data sets (of sizes ranging from hundreds of terabytes to hundreds of petabytes) across the data industry.

Over the years, since Netflix donated Iceberg to Apache, it has been adopted by several major tech companies, including Adobe, Airbnb, Apple, Google, LinkedIn, Snapshot and Snowflake.

Apache Iceberg and AI

As more organisations are navigating the uncharted sea called 'Enterprise AI', data management expert Cloudera is now adopting Apache Iceberg.

With the open data lakes created by Apache Iceberg, customers will get strong analytics and AI capabilities for their enterprise data, irrespective of whether the data is cloud based or on-premise. It is becoming a key technology for multi-vendor data ecosystems to get the most from AI. Cloudera recently claimed that its Iceberg-based solutions are managing 25 million terabytes of data.

With the increased support of Apache Iceberg, generative AI and large language models (LLMs), organisations can utilise data effectively. This further enables users to tap into more of their data and leverage it in different ways. END

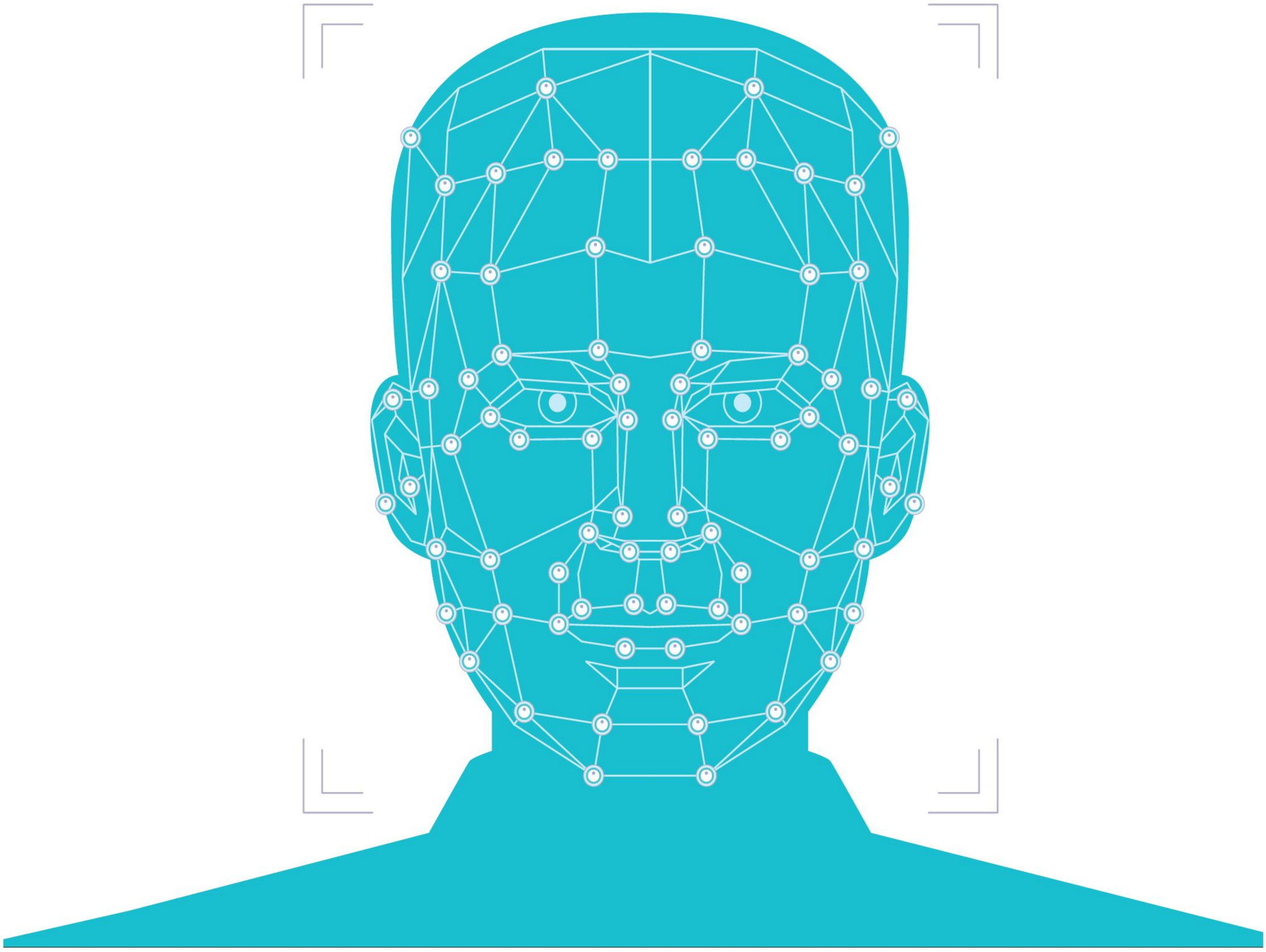
References

- Article 'Why Apache Iceberg will rule data in the cloud' by James Malone
- Blog 'Apache Iceberg: An architectural look under covers'; *dremio.com*
- Article 'Apache Iceberg becoming large scale analytics of choices' by David Curry
- Certain articles appeared in *Computer Weekly*

By: Vinayak R. Adkoli

The author holds a B.E. in industrial production and has served as a lecturer in three different polytechnics for 10 years. He is also a freelance writer and cartoonist.

OpenCV: Bringing AI Within Your Reach



AI technology is becoming more accessible than ever before, thanks to computer vision and IoT. OpenCV, an open source computer vision and machine learning software library, is helping small and big businesses embrace AI with ease.

Computer vision is a field of artificial intelligence (AI) that involves the use of algorithms to enable machines to interpret and understand images or videos from the real world.

With computer vision, machines can not only see but also understand the world around them like never before. It's incredible how this technology has made it possible for AI systems to recognise faces, objects, and even emotions! And it doesn't stop there! From healthcare to agriculture, computer vision is revolutionising the way we approach complex problems. With its integration with IoT, the possibilities are endless!

But why is this integration so critical? The answer is simple: accuracy. With computer vision, we can eliminate human error and ensure that the decisions we make are based on reliable data.

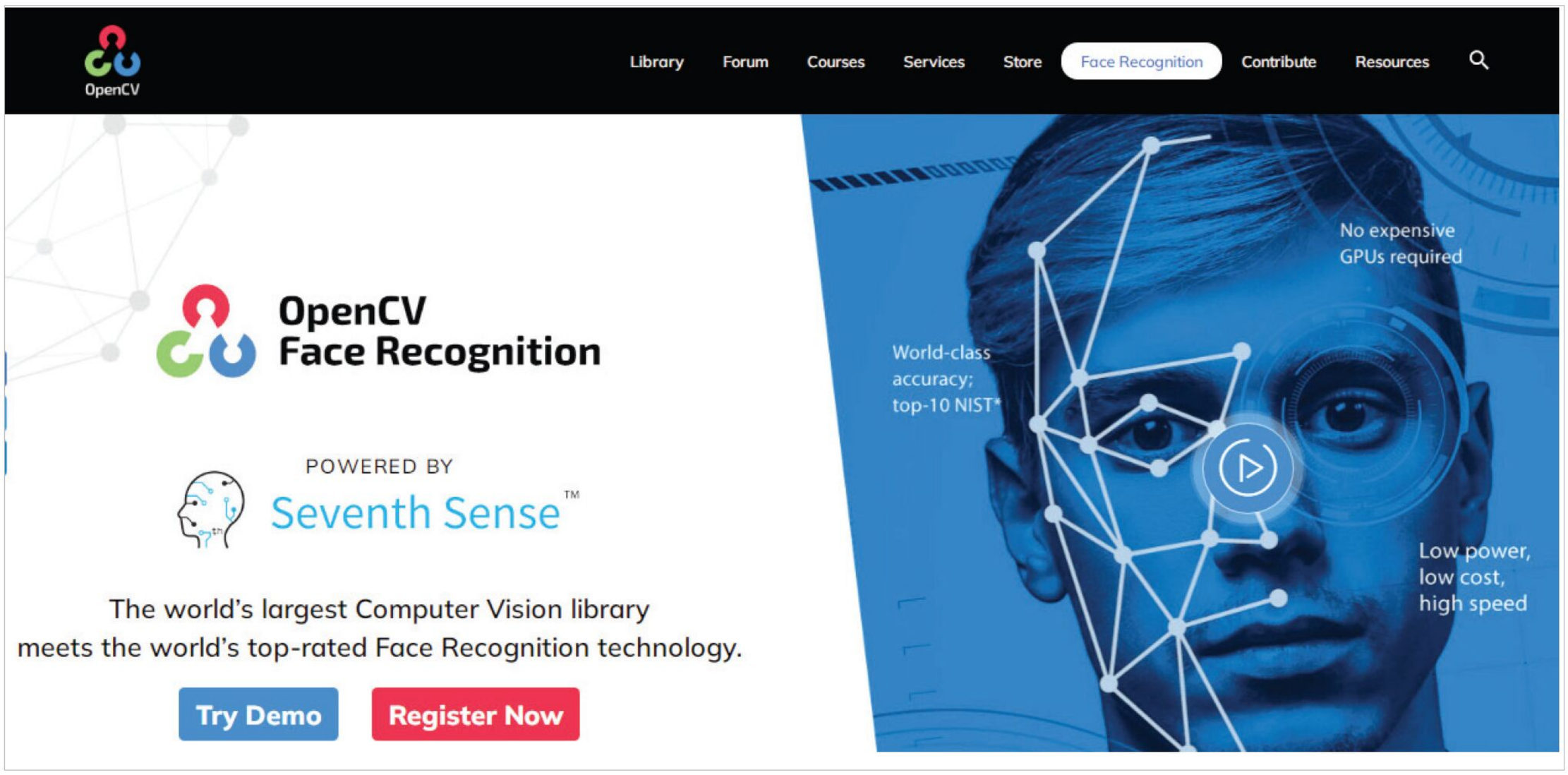


Figure 1: OpenCV library for computer vision

IoT meets computer vision for a wide range of applications

The integration of AI and Internet of Things (IoT) has opened up new avenues for the application of computer vision in a range of industries and use cases.

Intelligent spaces: One of the primary applications of computer vision in the context of AI and IoT is in the field of smart homes and buildings. With the integration of sensors and cameras, buildings can be equipped with systems that monitor and analyse data such as temperature, light, and occupancy. This information can be used to automate various processes such as adjusting the temperature or turning off lights when a room is unoccupied.

Healthcare: By integrating cameras and sensors into medical equipment and devices, healthcare providers can collect data and images to aid in diagnoses and treatment. For example, computer vision algorithms can analyse X-ray or MRI scans to detect early signs of disease or abnormalities.

Manufacturing: In the manufacturing industry, computer vision and AI can be used to optimise production processes and improve quality control. By integrating cameras and sensors into manufacturing equipment, manufacturers can collect data and images to identify defects and improve efficiency. Additionally, computer vision can be used to monitor employee performance and ensure compliance with safety regulations.

Defence industry: Computer vision, AI and IoT have also found several applications in the defence industry.

- *Surveillance and security:* Computer vision algorithms can be integrated with cameras and sensors to monitor sensitive areas and detect potential threats. The system can identify individuals and objects, and track their movements in real-time, enabling a quick response to any security breaches.
- *Autonomous systems:* Drones and unmanned vehicles equipped with computer vision and AI can be used for surveillance and reconnaissance, as well as for carrying out missions such as search and rescue operations or delivering supplies to remote areas.
- *Predictive maintenance:* In the defence industry, equipment downtime can have serious consequences. By integrating sensors with computer vision and AI, maintenance teams can monitor equipment performance in real-time and predict when maintenance is required, reducing downtime and ensuring readiness.

Agriculture: Computer vision can be used to monitor crop growth, detect plant diseases or pests, and optimise irrigation and fertilisation. This can improve crop yield and reduce waste.

Retail: Retailers can use computer vision and AI to analyse customer behaviour, such as how they interact with products and which products they prefer. This data can be used to improve store layout, inventory management, and personalised marketing strategies.

Traffic management: Integrating sensors and cameras with computer vision and AI can help monitor traffic flow, identify traffic patterns, and optimise traffic lights. This can reduce traffic congestion and improve road safety.

OpenCV: Encouraging innovation

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library that is widely used in various fields such as robotics, video analysis, and image processing.

One of its key features is the ability to process images and videos in real-time. It also offers a wide range of functions and algorithms for image processing, feature detection, object recognition, and more. As OpenCV is cross-platform, it can be used on various operating systems such as Windows, Linux, and macOS. The library is written in C++, but has interfaces for Python, Java, and other programming languages.

OpenCV's open source nature enables developers to customise and extend its functions to suit their specific needs. This makes it a popular choice for both researchers and developers in the field of computer vision.

Installation:

- Install Python on your system if it is not already installed.
- Install OpenCV using *pip* by running the following command in your terminal: *pip install opencv-python*.

Working with OpenCV:

- Import the OpenCV library in your Python code using the *import cv2* command.
- Load an image or a video file using the *cv2.imread()* or *cv2.VideoCapture()* functions, respectively.
- Perform operations such as resizing, cropping, filtering, or object detection using the various functions available in the OpenCV library.
- Display the result using the *cv2.imshow()* function.
- Wait for a key press using the *cv2.waitKey()* function.
- Release the resources and close the window using the *cv2.release()* and *cv2.destroyAllWindows()* functions, respectively.

Here is an example code snippet that demonstrates how to load an image using OpenCV, resize it, and display it:

```
import cv2
# Load the image
img = cv2.imread('image.jpg')
# Resize the image
resized_img = cv2.resize(img, (500, 500))
# Display the image
cv2.imshow('Image', resized_img)
# Wait for a key press
cv2.waitKey(0)
# Release the resources and close the window
cv2.release()
cv2.destroyAllWindows()
```

OpenCV provides a comprehensive collection of optimised algorithms for various computer vision and image processing tasks. With the above guide, you can start exploring and experimenting with OpenCV to develop your own computer vision applications.

Picturing the future: OpenCV's game-changing applications

OpenCV can be used in assorted high performance applications.

Object detection and recognition: Researchers can use OpenCV to develop algorithms that detect and recognise objects in images or videos. This could be applied to a variety of industries, including surveillance, automotive, and robotics.

Medical imaging: OpenCV can be used to analyse medical images such as X-rays, MRIs, and CT scans. Researchers can develop algorithms to detect and classify different types of diseases, improving the accuracy and efficiency of medical diagnoses.

Autonomous vehicles: This library can be used to develop computer vision systems for autonomous vehicles, enabling them to detect and respond to objects in their environment. This could lead to safer and more efficient transportation systems.

Augmented reality: It can be used to develop augmented reality applications, overlaying digital information onto real-world objects. This could be applied to gaming, education, and advertising.

Human-robot interaction: OpenCV can be used to develop computer vision systems that enable robots to interact with humans more effectively. This could be applied to a variety of industries, including healthcare, manufacturing, and services.

OpenCV also provides a broad scope for research and development across various fields. It can be extensively used in robotics, autonomous vehicles, medical imaging, and surveillance systems, among others. OpenCV's extensive range of features and algorithms, including image processing, object detection and tracking make it a popular choice for researchers and developers alike. Its open source nature provides opportunities for customisation, making it a powerful tool for developing computer vision applications tailored to specific needs.

OpenCV's libraries for AI and real-time scenarios

OpenCV provides a wide range of tools and functionalities for building computer vision applications. Here are some OpenCV-based open source libraries for AI and real-time scenarios.

- ***YOLOv4:*** YOLOv4 is an object detection algorithm that uses deep neural networks to detect and classify objects in real-time. It is based on the Darknet framework and uses OpenCV for image processing *(https://github.com/AlexeyAB/darknet)*.
- ***OpenPose:*** This is an open source library for real-time 2D and 3D pose detection. It uses deep learning

and computer vision techniques to estimate the poses of multiple people in real-time *(https://github.com/CMU-Perceptual-Computing-Lab/openpose)*.

- ***TensorFlow Object Detection API:*** This is an open source library for object detection and classification. It uses TensorFlow and OpenCV for training and inference of deep neural networks for object detection in real-time scenarios *(https://github.com/tensorflow/models/tree/master/research/object_detection)*.
- ***DLib:*** DLib is a modern C++ toolkit for building computer vision applications. It includes a wide range of algorithms and tools for object detection, face recognition, and more *(https://github.com/davisking/dlib)*.
- ***OpenCVSharp:*** This is a .NET wrapper for the OpenCV library. It allows developers to use OpenCV in their .NET applications.
- ***Emgu CV:*** A cross-platform .NET wrapper for the OpenCV library, it provides access to OpenCV functions from .NET languages such as C#, VB.NET, and F#.
- ***OpenFace:*** OpenFace is an open source facial behaviour analysis toolkit that uses OpenCV for facial landmark detection and tracking.
- ***Mahotas:*** A Python library for image processing and computer vision, it uses OpenCV for basic image processing operations.
- ***scikit-image:*** This Python library for image processing and computer vision uses OpenCV for some image processing functions.

These OpenCV-based open source libraries provide developers with a range of tools and functionalities for building AI and real-time computer vision applications.

Leading platforms for computer vision

If you are looking for similar options as the OpenCV platform, we have you covered. Here are some popular programming platforms and libraries for computer vision with their respective URLs.

- **TensorFlow:** TensorFlow is an open source platform for developing and deploying machine learning models. It includes a module for computer vision tasks such as image recognition, object detection, and segmentation *(https://www.tensorflow.org/)*.
- **PyTorch:** PyTorch is an open source machine learning framework that offers a range of tools and libraries for building and deploying deep learning models. It has a module for computer vision tasks such as image classification, object detection, and segmentation *(https://pytorch.org/)*.
- **Keras:** Keras is a high level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It provides a simple and intuitive way to build and train deep learning models for computer vision tasks such as image recognition and segmentation *(https://keras.io/)*.
- **MATLAB:** MATLAB is a programming language and numerical computing environment widely used in engineering and scientific applications. It includes a comprehensive toolbox for computer vision tasks such as image processing, computer vision system design, and machine learning *(https://www.mathworks.com/products/computer-vision.html)*.
- **SimpleCV:** SimpleCV is an open source framework for building computer vision applications in Python. It provides a high level interface for common CV tasks such as image processing, feature extraction, and object recognition *(https://simplecv.org/)*.
- **TorchVision:** TorchVision is a computer vision library that is part of the PyTorch ecosystem. It provides tools and models for image and video classification, object detection, and segmentation *(https://pytorch.org/vision/)*.
- **ImageJ:** ImageJ is open source image processing and analysis software written in Java. It includes a range of tools for image processing, segmentation, and feature extraction *(https://imagej.net/)*.

These platforms and libraries provide developers with a range of tools and functionalities to build and deploy computer vision applications.

Final thoughts

It's no secret that AI is the future. The synergy between computer vision and IoT can solve some of the world's most pressing issues. But as exciting as that sounds, the technology has remained quite exclusive to those who have the budget to back it up. That's where OpenCV comes in.

By democratising AI, platforms such as OpenCV are redefining scenarios that have traditionally been out of reach for smaller businesses. This shift not only promotes innovation but also opens doors for new talent and creates an opportunity for previously underserved markets to thrive.

Having access to powerful technology shouldn't be exclusive. OpenCV helps level the playing field for everyone. END

By: Dr Gaurav Kumar and Amit Doegar

Dr Gaurav Kumar is a associated with various academic and research institutes for delivering expert lectures and conducting technical workshops on the latest technologies and tools.

Amit Doegar is a associate professor in the Department of Computer Science and Engineering, National Institute of Technical Teachers Training & Research (NITTTR), Chandigarh.

Data Analysis:
Web Scripting with Python

Web scripting helps to retrieve and analyse data from a website efficiently. In this comprehensive guide to web scraping with Python web scripting, you will learn about how to use the *requests* and *BeautifulSoup* libraries to do so.



Image Source: https://www.freepik.com

With the rise of web-based services, it has become mandatory for us to orient ourselves to as many of them as possible — from social networking sites to daily utilities and financial services. As these web services have become rich and reliable sources of information, businesses are using them for their analytical exercises. Web scripting is playing a critical role in web scraping. This is an essential programming technology to analyse a web page and is used to obtain information from the websites. Retrieving data from a reliable website is a common exercise for many establishments. Copying, pasting, or downloading are not practical solutions for such a job. Moreover, sometimes it becomes mandatory to process that information at regular intervals for data analysis. Web scripting is required in these circumstances, to process and analyse that data efficiently.

One industry that extensively uses scrapped data is social media marketing. Companies in this space scrape and extract data from numerous websites to understand user sentiments, current trends/affairs, news, and feedback. This information helps to create effective marketing campaigns and business strategies.

In this article, we will explore how to perform web scraping using the *requests* library and the *BeautifulSoup* library of Python.

Python as a web scraper

A Python web scraper is a software or script to download the contents of multiple web pages and extract data from them. The functioning of a web scraper is shown in Figure 2.

Step 1: Downloading contents from web pages

In this step, a web scraper reads the requested page from a URL.

Step 2: Extracting data

Then the web scraper parses and extracts structured data from the requested contents.

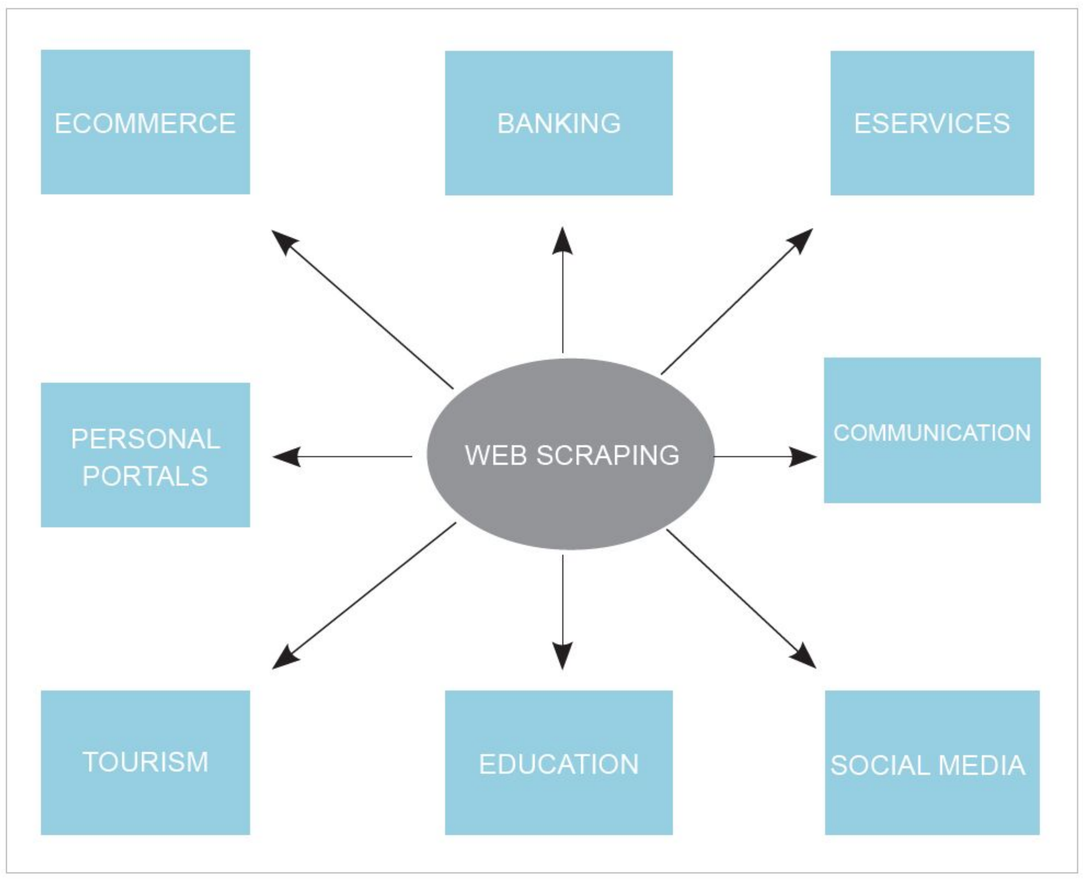


Figure 1: Web scraping as a service

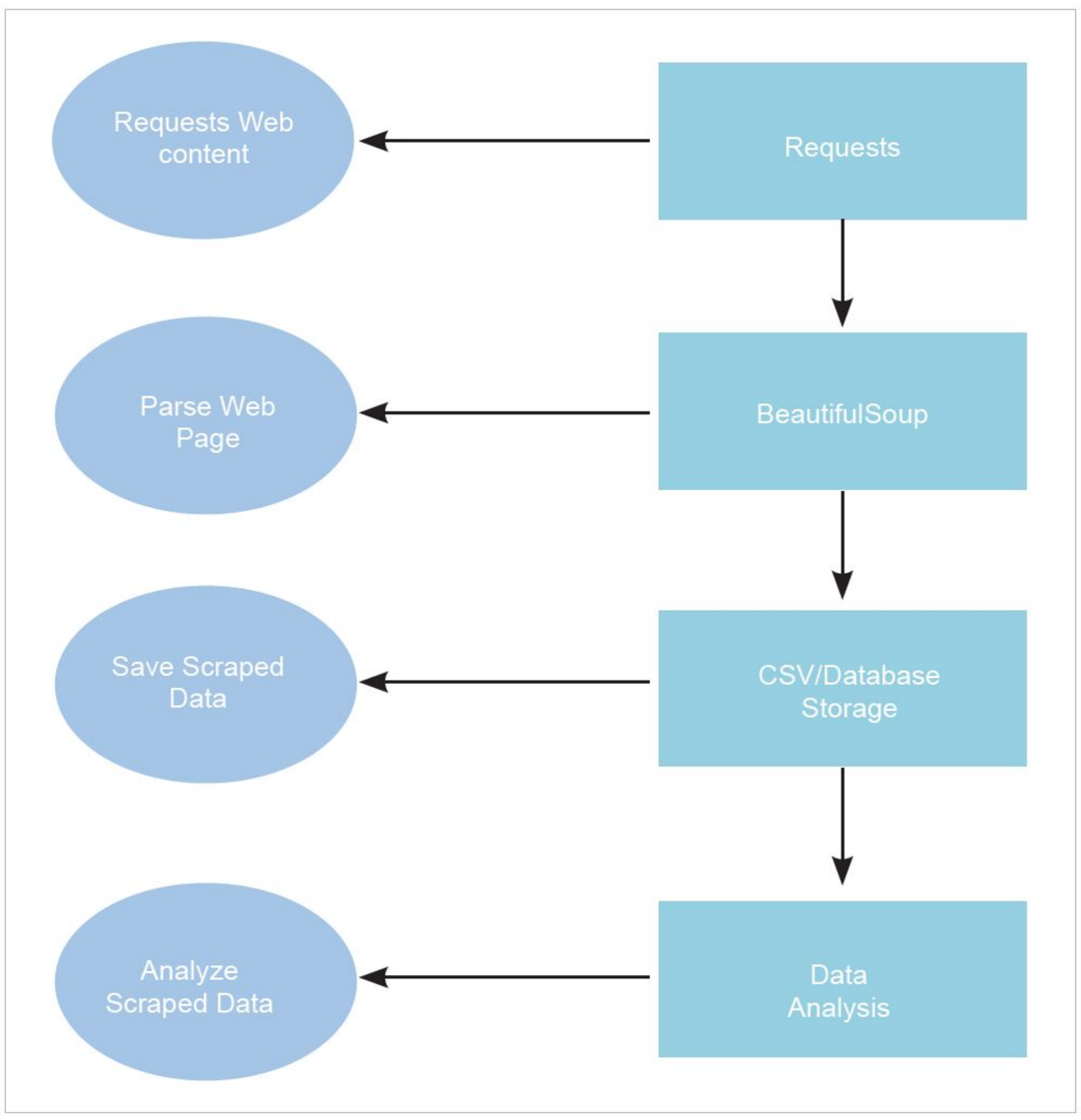


Figure 2: Schematic diagram of different stages of web scripting

Step 3: Storing the data

Subsequently, the web scraper stores the extracted data in CSV, JSON, or a database.

Step 4: Analysing the data

From these well-established formats, the web scraper initiates analysis of the stored data.

Web scraping of Python can be initiated with two simple library modules — *requests* and *BeautifulSoup*. In this article, we will demonstrate their functions as well as Python web scripting, with a few illustrations to help you understand the basics of web scraping.

requests module

Making HTTP requests and receiving responses are done with the help of the *requests* library. The management of the request and response is handled by built-in functionality in Python requests. Since this module is not shipped with core Python it is necessary to install it with *pip* separately.

```
pip install requests
```

Making a GET request

The Python *requests* module comes with some built-in ways to send GET, POST, PUT, PATCH, or HEAD requests to a given URI. An HTTP request may send data to a server or get it from a specified URI. It functions as a client-server request-response protocol. Information is retrieved from the specified server using the GET method and the given URI. The user data is encoded and added to the page request before being sent via the GET method.

```
import requests
# Making a GET request in Python
r = requests.get('https://www.tutorialspoint.com/python-for-
beginners-python-3/index.asp')
# check the status code for the response received
# success code - 200
print(r.status_code)
#print header
print(r.headers)
```

The output is:

```
200
{'Content-Encoding': 'gzip', 'Accept-Ranges': 'bytes',
'Access-Control-Allow-Credentials': 'true', 'Access-Control-
Allow-Headers': 'X-Requested
-With, Content-Type, Origin, Cache-Control, Pragma,
Authorization, Accept, Accept-Encoding', 'Access-Control-
Allow-Methods': 'GET, POST, PUT, P
ATCH, DELETE, OPTIONS', 'Access-Control-Allow-Origin': '*,
*;', 'Access-Control-Max-Age': '1000', 'Age': '515359',
'Cache-Control': 'max-age=25
92000', 'Content-Type': 'text/html; charset=UTF-8', 'Date':
'Thu, 25 May 2023 08:05:00 GMT', 'Expires': 'Sat, 24 Jun 2023
08:05:00 GMT',
Modified': 'Fri, 19 May 2023 08:55:43 GMT', 'Server': 'ECAcc
(blr/D141)', 'Vary': 'Accept-Encoding', 'X-Cache': 'HIT',
'X-Frame-Options': 'SAME
ORIGIN, SAMEORIGIN', 'X-Version': 'OCT-10 V1', 'X-XSS-
Protection': '1; mode=block', 'Content-Length': '14119'}
```

The response 'r' is a powerful object with a plethora of methods and features that aid in normalising data or offer the

finest code. For instance, one may assess whether or not the request was correctly processed using the status code provided by *r.status_code* and *r.header*.

BeautifulSoup library

The *BeautifulSoup* module is used to extract information from the HTML and XML files. It offers a parse tree together with tools for exploring, searching, and updating it. It generates a parse tree from the page's source code, which is used to organise data more logically and hierarchically.

The purpose of this module is to perform rapid reversing tasks like screen scraping. It is effective due to three aspects.

1. It provides some simple and minimal codes for analysing a text and extracting the required items from the text. It provides facilities for directing, finding, and altering a parse tree.
2. It automatically converts outputs to UTF-8 and receives records at Unicode.
3. Parsers like LXML and HTML are often used to experiment with various parsing strategies or execution performances for flexibility.

```
pip install beautifulsoup4
```

Since this module is not also shipped with core Python it is necessary to install it with *pip* separately.

Examining the target website

Inspecting the target website is required before developing a scripting program. One can do this by right-clicking on the target page and selecting *Developer tools*.

After clicking the *Developer tools* button of the open browser, one can inspect the internal codes of the open web page. We shall consider Chrome as the browser for this article.

The developer's tools allow a view of the site's document object model (DOM). From the display, one can examine the structure of the HTML coding of the page.

Parsing the HTML

After examining the scripting of the page, one can parse the raw HTML code into some useful information. For that, it's necessary to create a *BeautifulSoup* object by specifying the required parser. As this module is built on top of the HTML parsing libraries like *html5lib*, *lxml*, and *html.parser*, etc, all these libraries can be called along with this module. With this module, it's possible to parse a requested web page into its original HTML page. Here is an example:

```
import requests
from bs4 import BeautifulSoup

# GET request
r = requests.get('https://flask-bulma-css.appseed.us/')
# check the status code for the response received
print(r.status_code)

# Parsing the HTML
```

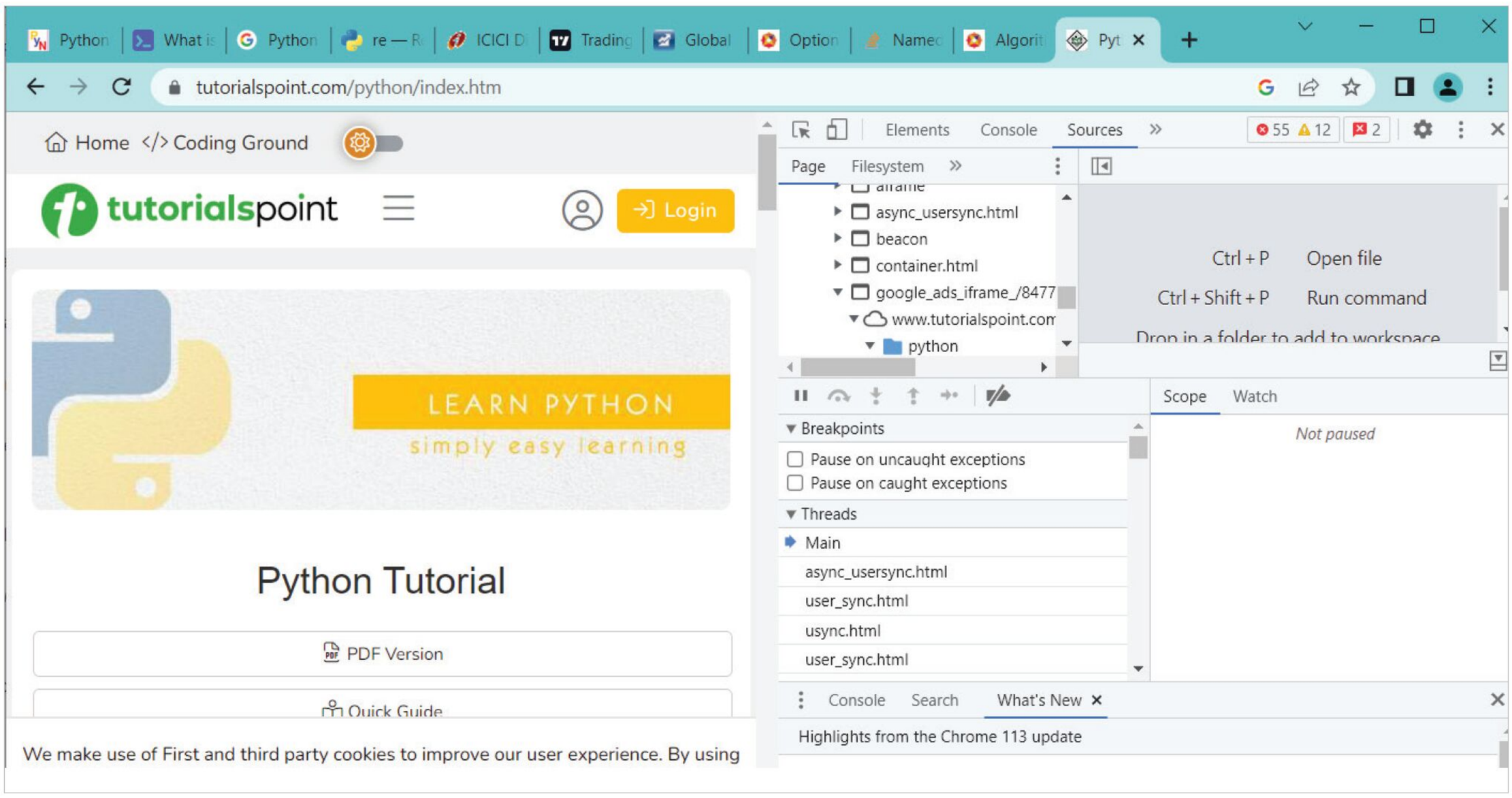


Figure 3: *Developer tool* in Google Chrome

```
htmlObj = BeautifulSoup(r.content, 'html.parser')
print(htmlObj.prettify())
```

The output is:

```
200
<!DOCTYPE html>
<html lang="en">
<head>
<!-- Required meta tags always come first -->
<meta charset="utf-8"/>
<meta content="width=device-width, initial-scale=1,
shrink-to-fit=no" name="viewport"/>
<meta content="ie=edge" http-equiv="x-ua-compatible"/>
<title>

Flask Bulma CSS - BulmaPlay Open-Source App | AppSeed App
Generator
```

For further analysis, it's possible to item-wise parse the HTML page into different parent and child classes.

```
import requests
from bs4 import BeautifulSoup
# Making a GET request
r = requests.get('https://flask-bulma-css.appseed.us/')

# Parsing the HTML
htmlobj = BeautifulSoup(r.content, 'html.parser')

# Getting the title tag
print(htmlobj.title)

# Getting the name of the tag
print(htmlobj.title.name)

# Getting the name of parent tag
print(htmlobj.title.parent.name)
# use the child attribute to get
# The name of the child tag
print(htmlobj.title.children.__class__)
```

The output is:

```
<title>Python Tutorial | Learn Python Programming</title>
title
head
<class 'list_iterator'>
```

Finding HTML page elements

Now, we want to extract some valuable information from the HTML text. The retrieved content in the *htmlobj* object may contain some nested structures. From these nested structures, different HTML elements can be parsed to retrieve valuable information from the web page.

HTML element search by class

We can see in Figure 4 that there is a *div* tag with the *class* tag entry. The *found* class method can be used to locate the tags. This

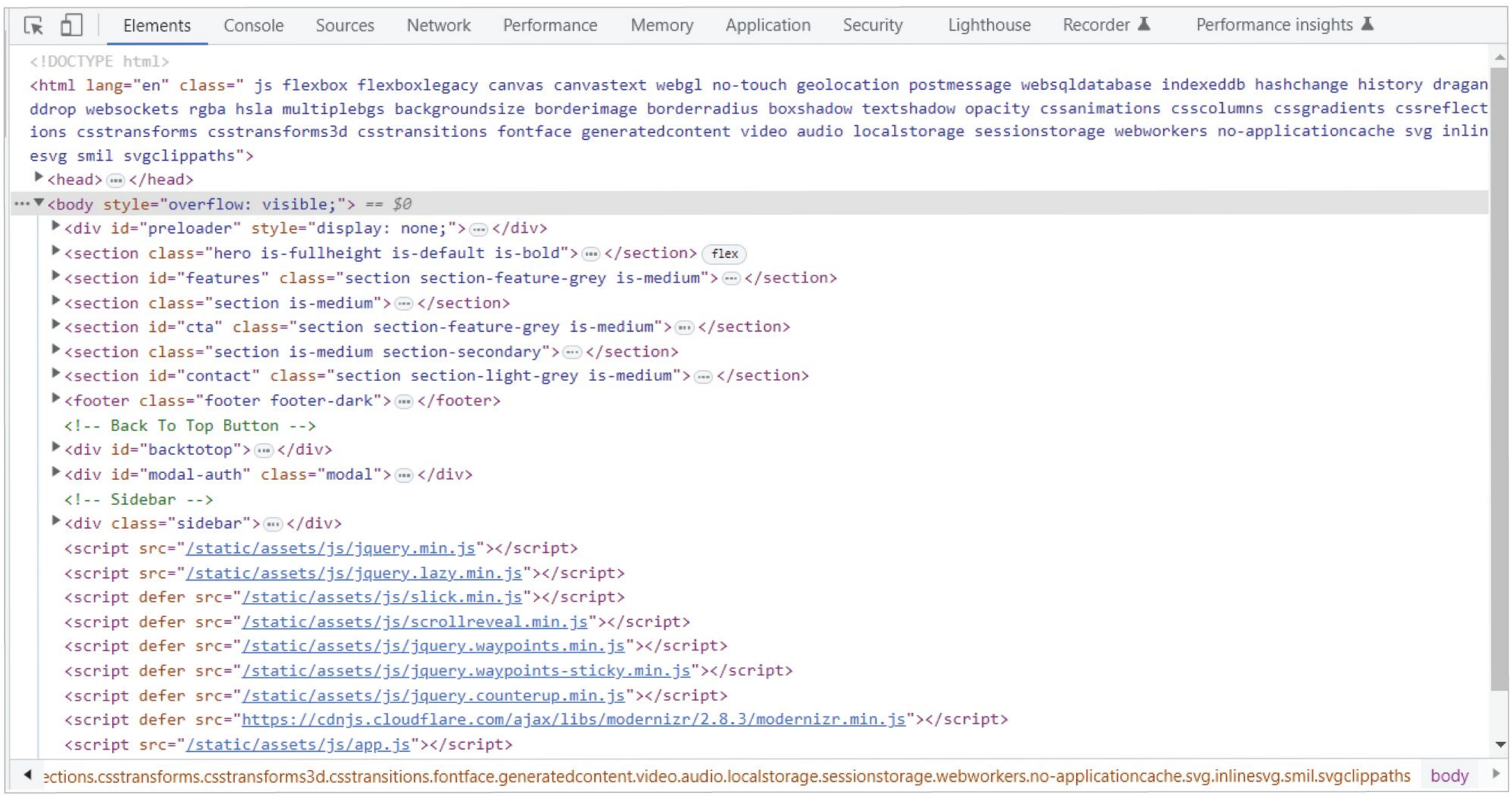


Figure 4: Inspection of web page script using a browser's developer tool

class searches for the specified tag with the specified property. In this instance, it will locate every *div* with the *class* entry content. Although we have all of the website's information, you can see that all the elements other than the <*a*> have been eliminated.

```
import requests
from bs4 import BeautifulSoup

# Making a GET request
r = requests.get('https://flask-bulma-css.appseed.us/')

# Parsing the HTML
htmlobj = BeautifulSoup(r.content, 'html.parser')

s = htmlobj.find('div', class_="sidebar")

content = s.find_all("a")

print(content)
```

The output is:

```
[<a class="sidebar-close" href="javascript:void(0);"><i
data-feather="x"></i></a>, <a href="#"><span class="fafa-
info-circle"></span>About</a>,<a href="https://github.com/
app-generator/flask-bulma-css" target="_blank">Sources</a>, <a
href="https://blog.appseed.us/bulmaplay-built-wit
h-flask-and-bulma-css/" target="_blank">Blog Article</a>,
<a href="#"><span class="fa fa-cog"></span>Support</a>, <a
href="https://discord.gg/f

ZC6hup" target="_blank">Discord</a>, <a href="https://
appseed.us/support" target="_blank">AppSeed</a>]
```

Identification of HTML elements

In the example above, we identified the components using the class name; now, let's look at how to identify items using their IDs. Let's begin this process by parsing the page's *cloned-navbar-menu* content. Examine the website to determine the tag of the '*cloned-navbar-menu*'. Then, finally, search all the *path* tags under the tag <*a*>.

```
import requests
from bs4 import BeautifulSoup

# Making a GET request
r = requests.get('https://flask-bulma-css.appseed.us/')

# Parsing the HTML
htmlObj = BeautifulSoup(r.content, 'html.parser')

# Finding by id
s = htmlObj.find('div', id="cloned-navbar-menu")

# Getting the "cloned-navbar-menu"
a = s.find('a' , class_="navbar-item is-hidden-mobile")

# All the paths under the above <a>
content = a.find_all('path')
print(content)
```

```
[<path class="path1" d="M 300 400 L 700 400 C 900 400 900 750
600 850 A 400 400 0 0 1 200 200 L 800 800"></p

ath>, <path class="path2" d="M 300 500 L 700 500"></path>,
<path class="path3" d="M 700 600 L 300 600 C 100
600 100 200 400 150 A 400 380 0 1 1 200 800 L 800 200"></
path>]
```

Iterating over a list of multiple URLs

To scrape many sites, each of the URLs need to be scraped individually, and it is required to hand-code a script for each such web page.

Simply create a list of these URLs and loop over it. It is not required to create code for each page to extract the titles of those pages; instead, just iterate over the list's components. You may achieve that by following the example code provided here.

```
import requests
from bs4 import BeautifulSoup as bs

URL = ['https://www.researchgate.net', 'https://www.linkedin.
com']

for url in range(1,2):
    req = requests.get(URL[url])
    htmlObj = bs(req.text, 'html.parser')

    titles = htmlObj.find_all('titles',attrs={'class','head'})
    if titles:
        for i in range(1,7):
            if url+1 > 1:
                print(titles[i].text)
    else:
        print(titles[i].text)
```

Storing web contents into a CSV file

After scraping the web pages, one can store the elements into a CSV file. Let's consider the storing of a tag-value dictionary list in a CSV file. The dictionary elements will be written to a CSV file using the CSV module. Check out the example below for a better understanding.

Continued to page...82

Building Fast and Feature-Rich Web Applications Using HyperApp



In the world of web development, where speed and efficiency are paramount, developers are constantly on the lookout for frameworks that offer a lightweight yet powerful approach. HyperApp, a JavaScript framework, has gained popularity for its minimalistic design and efficient functionality. In this article, we will explore its features, examine its benefits, and provide real-world examples.

HyperApp has emerged as a modern JavaScript library that empowers developers to build fast and feature-rich applications in the browser. With its minimalist design and focus on performance, HyperApp offers a compelling alternative for developers seeking a lightweight yet powerful framework.

What is HyperApp?

HyperApp is a front-end JavaScript framework that focuses on simplicity and performance. It follows a functional programming paradigm and draws inspiration from React, Redux, and Elm. With its tiny footprint (just1KB gzipped), HyperApp offers an ideal solution for building fast and responsive web applications.

This minimalistic JavaScript library weighs just a few kilobytes, making it ideal for building lightweight applications. It follows a unidirectional data flow architecture, similar to other popular frameworks like Redux, to ensure predictable state management. Its small size and efficient design translate into faster load times and improved performance for web applications.

To get started with HyperApp, we need to set up the development environment and create a basic project structure following the steps mentioned here:

1. Install HyperApp using npm or Yarn (use Python SDK as needed).
2. Create a new HyperApp project using the CLI or manually set up the project structure.

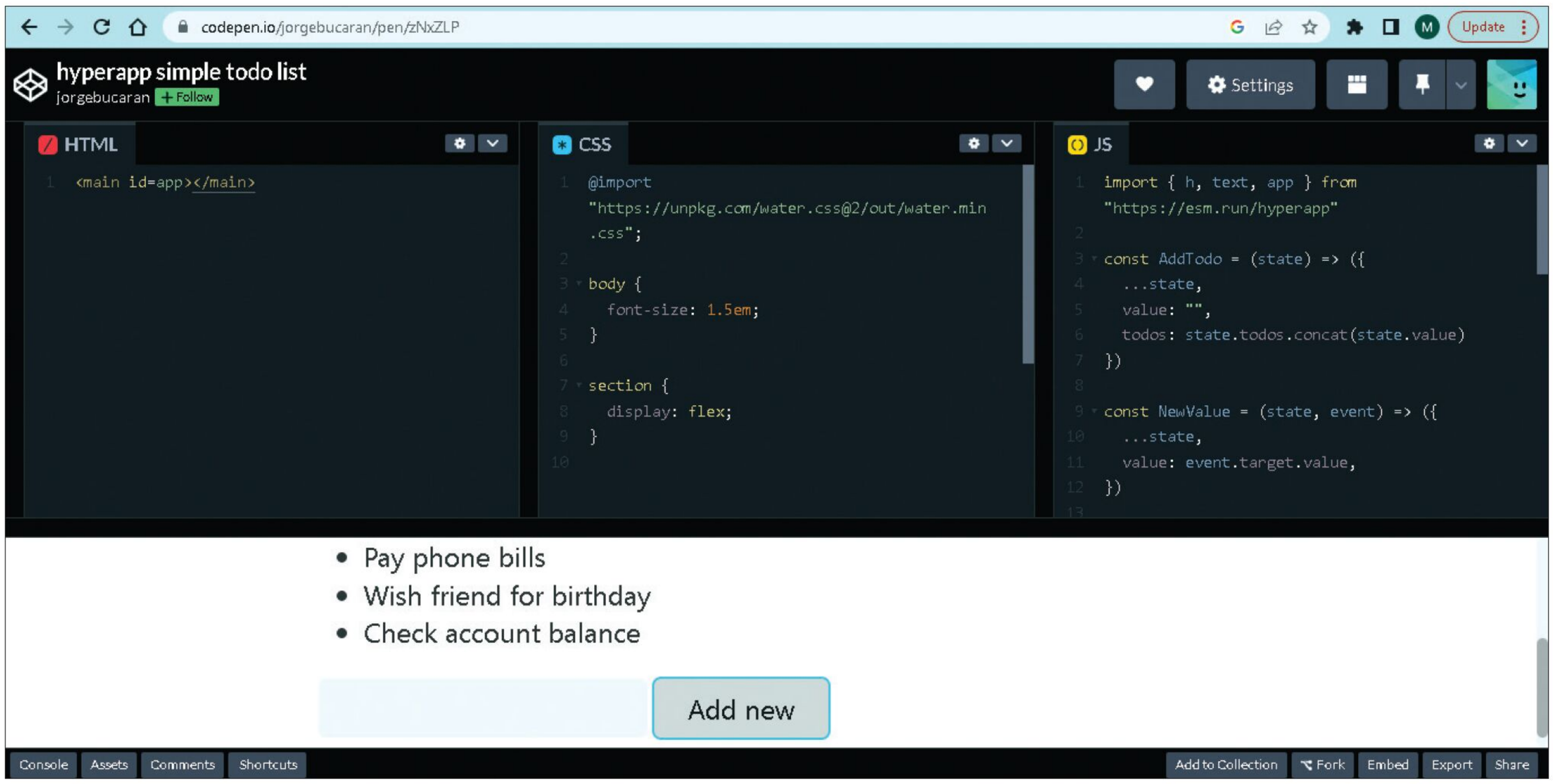


Figure 1: Screenshot of developing a HyperApp application in CodePen editor

3. Set up a basic HyperApp component using JSX syntax.
4. Render the component to the DOM.

HyperApp leverages JSX syntax, allowing developers to write expressive and declarative UI components. It boasts a simple and intuitive API, making it easy for developers to grasp and leverage its capabilities. The library employs a declarative approach, where developers describe the desired user interface state and let HyperApp handle the rendering and updates. This approach promotes code clarity and maintainability, allowing developers to focus on building application logic rather than worrying about the intricacies of the underlying framework.

HyperApp leverages a virtual DOM (VDOM) implementation to efficiently update and render UI components. By comparing the VDOM with the actual DOM, HyperApp minimises unnecessary re-renders, resulting in optimal performance. The library's efficient Diffing algorithm ensures that only the necessary updates are applied, reducing rendering overhead and enabling fast and responsive user interfaces.

Reactive state management

HyperApp incorporates a reactive state management system that simplifies data handling within applications. It provides a central store, inspired by Redux, where the application state is managed. Any changes to the state trigger automatic re-rendering of relevant components, ensuring a consistent and up-to-date UI. HyperApp's reactive approach allows for easier state management and promotes the development of scalable and maintainable applications.

Architecture of HyperApp

HyperApp embraces a modular architecture, allowing developers to selectively include features and extensions based on their project requirements. The library provides a rich ecosystem of plugins and community-contributed packages, offering additional functionalities such as routing, HTTP requests, and form handling. This modular approach promotes code reusability and empowers developers to customise the development environment based on their specific needs.

With its lightweight design and efficient rendering mechanism, HyperApp delivers exceptional performance, even in complex applications. The library's focus on minimalism and performance optimisation enables faster page loads, smoother transitions, and improved user experiences. HyperApp's scalability is evident in its ability to handle large data sets and complex application logic while maintaining optimal performance.

HyperApp benefits from an active and growing community that contributes to its ecosystem with plugins, tools, and resources. Its community-driven nature ensures ongoing support, updates, and documentation, making it easier for developers to get started and collaborate.

With HyperApp, developers can create responsive and scalable applications while maintaining code clarity and simplicity. As the ecosystem continues to grow and evolve, HyperApp is poised to play an increasingly significant role in the development of performant web applications.

Feature	HyperApp	React	Vue.js	Angular
Size	Lightweight (~7KB)	Larger (~100KB)	Larger (~80KB)	Larger (~500KB)
Virtual DOM	Yes	Yes	Yes	Yes
Performance	Fast	Fast	Fast	Fast
Reactivity	Reactive state	Reactive state	Reactive state	Two-way binding
Learning curve	Easy	Moderate	Moderate	Steep
State management	Built-in	Redux (external)	Vuex (external)	Built-in
Component reusability	High	High	High	High
Community support	Growing	Vast	Vast	Vast
Extensibility	Modular	Plugin ecosystem	Plugin ecosystem	Module-based
Mobile support	Yes	Yes	Yes	Yes
Documentation	Well-documented	Comprehensive	Comprehensive	Comprehensive

HyperApp compared with other popular JS frameworks

Building a user interface with HyperApp

This lightweight JavaScript library for building user interfaces follows a declarative approach similar to React but with a much smaller footprint. Here's a step-by-step guide on how to build a user interface with HyperApp.

Step 1: Set up a new project

Start by creating a new HTML file and include the HyperApp library. You can either download the library and reference it locally or use a content delivery network (CDN) to include it in your project. Here's an example using the CDN:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>HyperApp UI</title>
<script src="https://unpkg.com/
hyperapp"></script>
</head>
<body>
<div id="app"></div>
<script src="app.js"></script>
</body>
</html>
```

Step 2: Define the initial state

In your JavaScript file (e.g., app.js), start by defining the initial state of your application. The state represents the data that will drive your UI. For example:

```
const state = {
};
count: 0
```

Step 3: Define actions

Actions in HyperApp are responsible for updating the state. They receive the current state as the first argument and can return a new state. Here's an example of an action that increments the count:

```
const actions = {
increment: () => (state) => ({
count: state.count + 1 })
};
```

Step 4: Define the view

The view describes the UI components and their structure. It's a function that returns a virtual DOM representation using HyperApp's JSX-like syntax.

```
const view  = (state, actions) => (
<div>
<h1>Count: {state.count}</h1>
<button onclick={actions.
increment)>Increment</button>
</div>
);
```

Step 5: Mount the application

Finally, mount the application by calling the 'app' function from HyperApp, passing the *state*, *actions*, and *view* as arguments. Here's an example:

```
const main = app(state, actions, view,
document.getElementById('app'));
```

That's it! A basic HyperApp application with a simple UI has been developed. Whenever the 'Increment' button is clicked, it will trigger the increment action, updating the *state* and re-rendering the *view*.

API request handling with HyperApp applications

Actions, effects, and asynchronous operations are frequently used in combination to handle API requests in HyperApp applications.

The following steps and source codes elaborate how to manage API calls in a HyperApp application.

Step 1: Define the initial state
Start by defining the initial state of your application, including properties related to API data. For example:

```
const state = {
todos: [ ],
isLoading: false,
};
```

Step 2: Define actions for API requests
Create actions that handle API requests. Typically, these actions will trigger asynchronous operations and update the *state* accordingly. Here's an example of an action that fetches to-dos from an API:

```
const actions = {
fetchTodos: () => async (state,
actions) => {
actions.setLoading (true);
actions.setError(null);
try {
const response = await fetch('https://
api.example.com/todos');
const todos = await response.json();
actions.setTodos (todos);
} catch (error) {
actions.setError('Failed to fetch
todos');
console.error('Failed to fetch todos:',
error);
}
actions.setLoading (false);
},
setTodos: (todos) => ({ todos }),
setLoading: (isLoading) => ({ isLoading
}),
setError: (error) => ({ error }),
};
```

In the above example, *fetchTodos* is an asynchronous action that sets the *isLoading* state to true, performs the API request using *fetch*, and updates the *todos* state with the fetched data. If an error occurs during the request, it is logged to the console.

Step 3: Define the view
Update your *view* function to conditionally render components based on the *state*. For example, you can show a loading indicator when the API request is in progress. Here's an updated example:

```
const view = (state, actions) => (
<div>
{state.isLoading ? (
<div>Loading...</div>
) : (
<div>
<ul>
{state.todos.map((todo) => (
<li>{todo.title}</li>
))}
</ul>
<button onclick={actions.
fetchTodos}>Fetch Todos</button>
</div>
    )}
</div>
);
```

Step 4: Mount the application
Mount the application as before, using the updated *state*, *actions*, and *view* functions:

```
const main = app(state,
actions, view, document.
getElementById('app'));
```

Now, when the 'Fetch Todos' button is clicked, it will trigger the *fetchTodos* action, which will make the API request, update the state with the fetched to-dos, and re-render the view.

We can extend this example by adding more actions for different API requests, handling request parameters, error handling, and more. Additionally, you might want to consider using HyperApp effects, such as the http effect, or external libraries like Axios for more advanced API handling.

Testing and debugging HyperApp applications

Testing and debugging are crucial steps in the development process of HyperApp applications to ensure that the application functions correctly and handles potential issues gracefully.

Here are some approaches and tools you can use for testing and debugging HyperApp applications.

Unit testing: Write unit tests to verify the behaviour of individual components, actions, or utility functions in isolation. Tools like Jest or Mocha can be used for testing HyperApp applications. Unit tests help identify and fix bugs at an early stage and ensure that changes to the codebase don't introduce regressions.

Integration testing: Perform integration tests to validate the interaction between different components, actions, and the overall application behaviour. Integration tests can be written using testing frameworks like Cypress or Selenium, which simulate user interactions and verify the application's functionality across multiple components.

Debugging tools: HyperApp provides debugging tools that can help identify and fix issues. For example, you can use the Redux DevTools extension for debugging state changes, actions, and their payloads. HyperApp also has a built-in *app({ ... })* method for logging state changes and rendering performance optimisations.

Console logging: Insert *console.log* statements at critical points in your code to log values, state changes, or function outputs during runtime. Console logging can help track the flow of execution and identify unexpected behaviour or incorrect data.

Error handling: Implement proper error handling and error messages to make debugging easier. Use try/catch blocks around critical sections of code, especially asynchronous operations and API requests. Log or display meaningful error messages to assist in identifying and resolving issues.

Browser developer tools: Leverage the browser's developer tools, such as the JavaScript console, network tab, and debugger. These tools allow you to inspect network requests, log JavaScript errors, step through code execution, and examine the state and structure of the application's DOM.

Code review: Have peers or team members review your code. Code reviews can help identify potential issues, provide valuable feedback, and ensure code quality.

User testing: Involve real users or testers in the process to gather feedback and identify potential issues from a user's perspective. User testing can uncover usability problems, edge cases, or unexpected behaviour that may not have been detected during development.

By combining these testing and debugging approaches, developers can ensure the reliability, functionality, and user experience of their HyperApp applications. END

References

- *hhttps://www.digitalocean.com/community/tutorials/js-hyperapp*
- *https://fireflysemantics.medium.com/what-is-reactive-state-management-51dcef222023*

By: Dr Magesh Kasthuri and Dr Anand Nayyar

Dr Magesh Kasthuri is a senior distinguished member of the technical staff and principal consultant at Wipro Ltd. This article expresses his views and not that of Wipro.

Dr Anand Nayyar is a PhD in wireless sensor networks and swarm intelligence. He works at Duy Tan University, Vietnam, and loves to explore open source technologies, IoT, cloud computing, deep learning and cyber security.

Continued to page...87

Example: Saving data from Python's *BeautifulSoup* to a CSV file:

```
import re
import requests
from bs4 import BeautifulSoup as bs
import csv

# Making a GET request
r = requests.get('https://flask-bulma-css.appseed.us/')

# Parsing the HTML
htmlObj = bs(r.content, 'html.parser')

# Finding by id
s = htmlObj.find('div', id="cloned-navbar-menu")

# Getting the leftbar
a = s.find('a' , class_="navbar-item is-hidden-mobile")

# All the li under the above ul
content = a.find_all('path')
lst = []
for i in range(3):
    dict = {}
    s = str(content[i])
    dict['path_name'] = re.search('class="path[1-3]"', s)
    dict['path_value'] = re.search('d=.*>$', s)
    lst.append(dict)
filename = 'page.csv'
with open(filename, 'w', newline='') as f:
    w = csv.DictWriter(f,['path_name','path_value'])
    w.writeheader()
    w.writerows(lst)
f.close()
```

The output CSV file is given below.

path_name	path_value
<re.Match object; span=(6, 19), match='class="path1"'>	<re.Match object; span=(20, 111), match='d="M 300 400 L 700 400 C 900 400 900 750 600 850 >
<re.Match object; span=(6, 19), match='class="path2"'>	<re.Match object; span=(20, 51), match='d="M 300 500 L 700 500"></path>'>
<re.Match object; span=(6, 19), match='class="path3"'>	<re.Match object; span=(20, 111), match='d="M 700 600 L 300 600 C 100 600 100 200 400 150 >

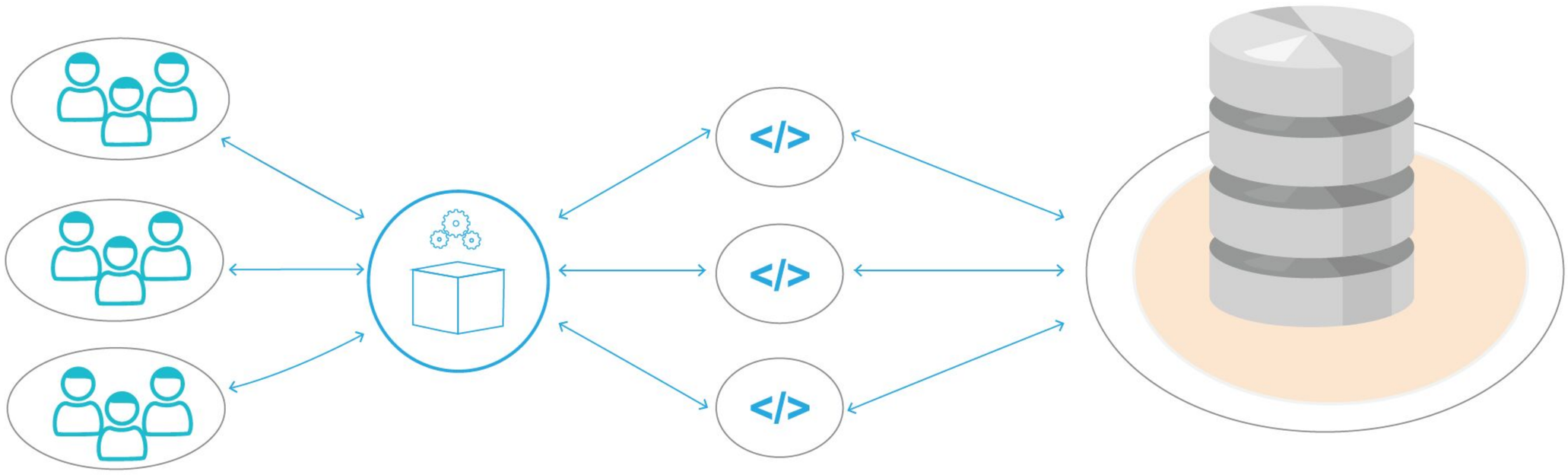
Web scraping with Python web scripting is a powerful tool for web data analysis. Here we have mentioned a few useful *requests* and *BeautifulSoup* library methods. Hope you will find the information useful and start your journey into this prevailing technology for data analytics of web information. END

By: Dr Dipankar Ray

The author has more than 20 years of experience in software development. He has published two books on MATLAB and medical image processing (published by PHI Learning and SPD).

Demystifying Connection Pools: A Deep Dive

Connection pooling can be a game changer for your app's performance. This final part of the two-part series of articles on connection pooling continues to uncover its magic.



In the first part of this series, we covered how typical connections are established, what connections pools are and how they work. We also started implementing a connection pool of our own, and figured out the basic building blocks needed to do so. Let's continue from where we left off.

We have already implemented the Get, Add and Close methods for our connection pool. Now, we need to tie them together and provide an interface to create a connection pool. We will implement a NewConnectionPool function as a constructor for a new connection pool. It takes the driver, dataSource, and maxSize arguments and returns a pointer to a new ConnectionPool instance. It first checks if the provided driver and dataSource arguments are valid by opening a connection to the database. If the connection is successful, it initialises a new connection pool with the provided maxSize argument. It then creates a new channel of *sql.DB objects and pre-populates it with maxSize database connections by creating a new database connection for each iteration of a loop. Finally, it returns the new ConnectionPool instance.

```
func NewConnectionPool(driver, dataSource string, maxSize
int) (*ConnectionPool, error) {

    // Validate driver and data source
    _, err := sql.Open(driver, dataSource)
    if err != nil {
        return nil, err
    }

    cp := &ConnectionPool{
        queue:       make(chan *sql.DB, maxSize),
        maxSize:     maxSize,
        currentSize: 0,
    }

    cp.isNotEmpty = sync.NewCond(&cp.lock)
    cp.isNotFull = sync.NewCond(&cp.lock)

    for i := 0; i < maxSize; i++ {
        conn, err := sql.Open(driver, dataSource)
        if err != nil {
            return nil, err
        }
        cp.queue <- conn
        cp.currentSize++
    }

    return cp, nil
}
```

Putting it all together

This is what our final custom ConnectionPool implementation looks like:

```
package pool

import (
    "database/sql"
```

```
    "errors"
    "fmt"
    "sync"
    _ "github.com/mattn/go-sqlite3"
)
type ConnectionPool struct {
    queue       chan *sql.DB
    maxSize     int
    currentSize int
    lock        sync.Mutex
    isNotFull   *sync.Cond
    isNotEmpty  *sync.Cond
}

func (cp *ConnectionPool) Get() (*sql.DB, error) {
    cp.lock.Lock()
    defer cp.lock.Unlock()

    // If queue is empty, wait
    for cp.currentSize == 0 {
        fmt.Println("Waiting for connection to be added back
in the pool")
        cp.isNotEmpty.Wait()
    }
    fmt.Println("Got connection!! Releasing")
    db := <-cp.queue
    cp.currentSize--
    cp.isNotFull.Signal()

    err := db.Ping()
    if err != nil {
        return nil, err
    }

    return db, nil
}

func (cp *ConnectionPool) Add(db *sql.DB) error {
    if db == nil {
        return errors.New("database not yet initiated. Please
create a new connection pool")
    }

    cp.lock.Lock()
    defer cp.lock.Unlock()

    for cp.currentSize == cp.maxSize {
        fmt.Println("Waiting for connection to be released")
        cp.isNotFull.Wait()
    }

    cp.queue <- db
    cp.currentSize++
    cp.isNotEmpty.Signal()

    return nil
}

func (cp *ConnectionPool) Close() {
    cp.lock.Lock()
    defer cp.lock.Unlock()

    for cp.currentSize > 0 {
        db := <-cp.queue
        db.Close()
        cp.currentSize--
        cp.isNotFull.Signal()
    }

    close(cp.queue)
}

func NewConnectionPool(driver, dataSource string, maxSize
int) (*ConnectionPool, error) {

    // Validate driver and data source
    _, err := sql.Open(driver, dataSource)
    if err != nil {
        return nil, err
    }

    cp := &ConnectionPool{
        queue:       make(chan *sql.DB, maxSize),
        maxSize:     maxSize,
        currentSize: 0,
    }

    cp.isNotEmpty = sync.NewCond(&cp.lock)
    cp.isNotFull = sync.NewCond(&cp.lock)

    for i := 0; i < maxSize; i++ {
        conn, err := sql.Open(driver, dataSource)
        if err != nil {
            return nil, err
        }
        cp.queue <- conn
        cp.currentSize++
    }

    return cp, nil
}
```

Of course, there are many ways in which this implementation can be improved upon. Typically, you can use

any variation of the bounded-queue to implement your own connection pool. Most connection pool implementations use bounded-queue (*https://www.oreilly.com/library/view/design-patterns-and/9781786463593/2ff33f7c-aab8-4a4d-bacc-c475c3d1c928.xhtml*) as the underlying data structure.

The complete implementation along with its usage is open sourced at *https://github.com/pratikms/dbconnectionpool* in case you wish to play around. I'll suggest running it in debug mode to watch the signalling magic of sync.Cond unfold.

Common connection pooling issues

While connection pooling can bring many benefits to an application, it is not without its challenges. Here are some common issues that can arise with connection pooling.

- ***Overuse of connection pools*:** Connection pools should be used judiciously, as overuse can result in a fall in the application's performance. This is because the connection pool itself can become a bottleneck if too many connections are being opened and closed, causing delays in database transactions.
- ***Pool size configuration errors*:** Connection pool size is an important consideration when implementing connection pooling. If the pool size is too small, there may not be enough connections available to handle peak traffic, resulting in errors or delays. On the other hand, if the pool size is too large, it can lead to unnecessary resource consumption and potential performance issues.
- ***Connection leaks*:** Connection leaks occur when a connection is not properly closed and returned to the pool after it has been used. This can lead to resource exhaustion, as unused connections will remain open and tie up valuable resources. Over time, this can result in degraded application performance and, in extreme cases, cause the application to crash.

To avoid these issues, it is important to monitor connection pool usage and performance regularly. Best practices such as setting appropriate pool size, tuning timeout and idle settings, and configuring automatic leak detection and recovery can help minimise the impact of these issues. Additionally, logging and alerting mechanisms can be put in place to help identify and remediate any issues that do occur.

Connection pooling in cloud environments

Connection pooling is an important consideration when designing applications for the cloud. Cloud environments offer several unique challenges, such as elastic scalability and dynamic resource allocation. Connection pooling can help address some of these challenges, but there are additional considerations to take into account.

In a cloud environment, applications may be running on multiple instances or virtual machines. This means that a single connection pool may not be sufficient to handle the load from all of these instances. Instead, it may be necessary to implement multiple connection pools, each handling a subset of the total workload.

Another consideration is the dynamic nature of cloud environments. Instances can be added or removed from the environment at any time, which means that the size of the connection pool may need to be adjusted accordingly. This can be achieved through automation tools or by implementing dynamic scaling rules based on metrics such as CPU usage or network traffic.

Security is also an important consideration when implementing connection pooling in the cloud. In a shared environment, it is important to ensure that connections are secure and cannot be accessed by unauthorised parties. This may involve implementing encryption or access control measures, such as IP filtering.

Finally, it is important to ensure that connection pooling is properly configured for the specific cloud environment being used. Each cloud provider may have its own specific requirements and recommendations for connection pooling, such as maximum pool size or connection timeouts. It is important to consult the provider's documentation and best practice guides to ensure that connection pooling is properly configured for optimal performance and reliability.

Connection pooling can be a valuable tool for optimising performance and managing resources in cloud environments. However, there are additional considerations that must be taken into account to ensure that connection pooling is properly implemented and configured for the specific cloud environment being used.

Final thoughts

In conclusion, connection pooling is a crucial concept in modern software development. It can help to improve application performance and scalability while reducing resource usage. By caching and reusing database connections, connection pooling can reduce the overhead of creating and destroying connections, leading to faster application response times and increased throughput.

However, connection pooling is not a silver bullet and must be used carefully and thoughtfully.

By understanding how connection pooling works, common issues to look out for, and best practices for implementation, software engineers can harness the power of connection pooling to build more performant, scalable, and reliable applications. END

By: Pratik Shivaraikar

The author is an open source software enthusiast, evangelist and contributor with a varied experience of working in the storage, security and wireless domains. He actively works on Go, Python, Node.js, PHP, and more.

Building a Chatbot Application with IBM Watson

The market for chatbots is expanding and is expected to continue growing. Did you know that IBM Watson can help build intelligent and efficient AI-based chatbot solutions?

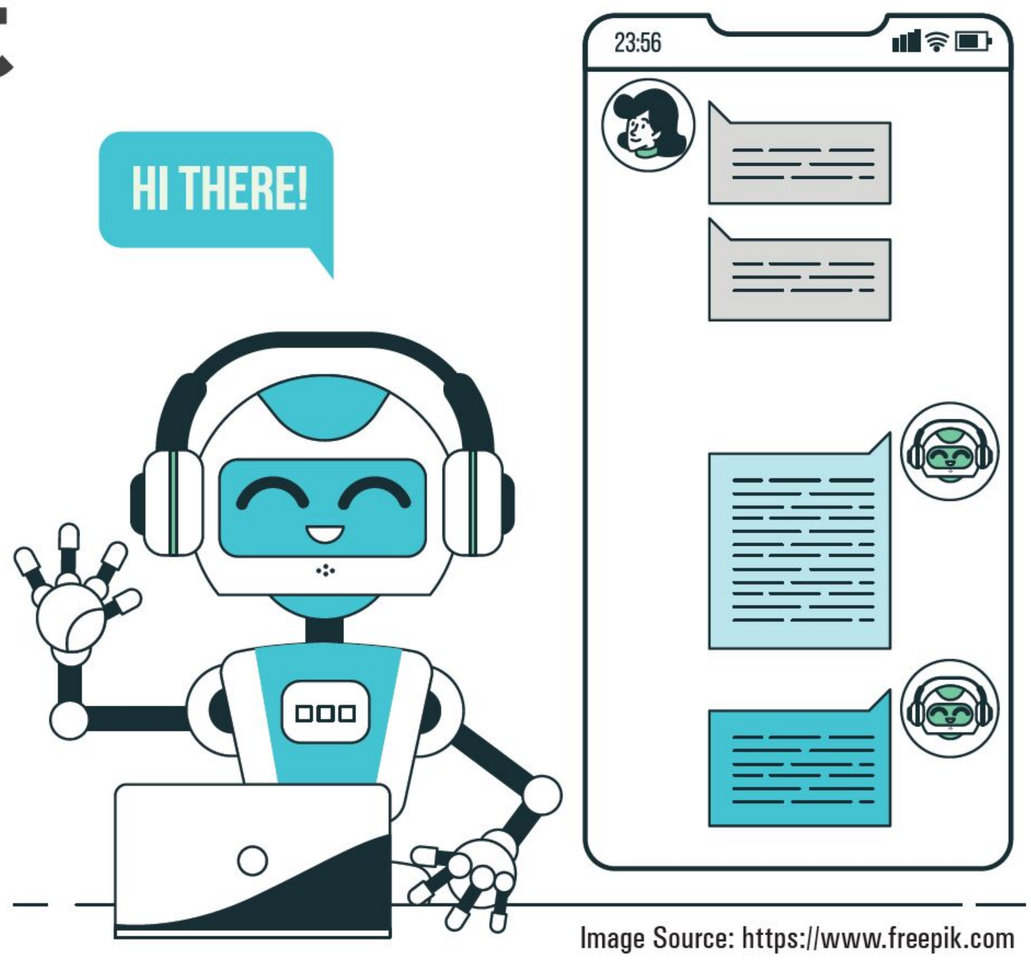


Image Source: https://www.freepik.com

AI-powered chatbots provide a more human-like experience, can carry on a natural conversation, and continuously improve over time. Their popularity stems from the improvements they bring to customer experiences, their 24/7 availability, cost-effective customer support, enhanced operational efficiency, easy scalability, support for multiple channels, data collection and analysis capabilities, continuous learning, and task automation. These benefits make them valuable tools for businesses across various industries. The working of an AI-based chatbot is demonstrated in Figure 1. As shown, the chatbot evaluates the users' requests, ascertains their intent, determines what they require, and then automatically generates and replies to them with the most relevant response.

Top industries using AI-based chatbots for their business

Various industries widely adopt AI-based chatbots to enhance customer engagement, improve operational efficiency, and provide personalised assistance. For example, chatbots

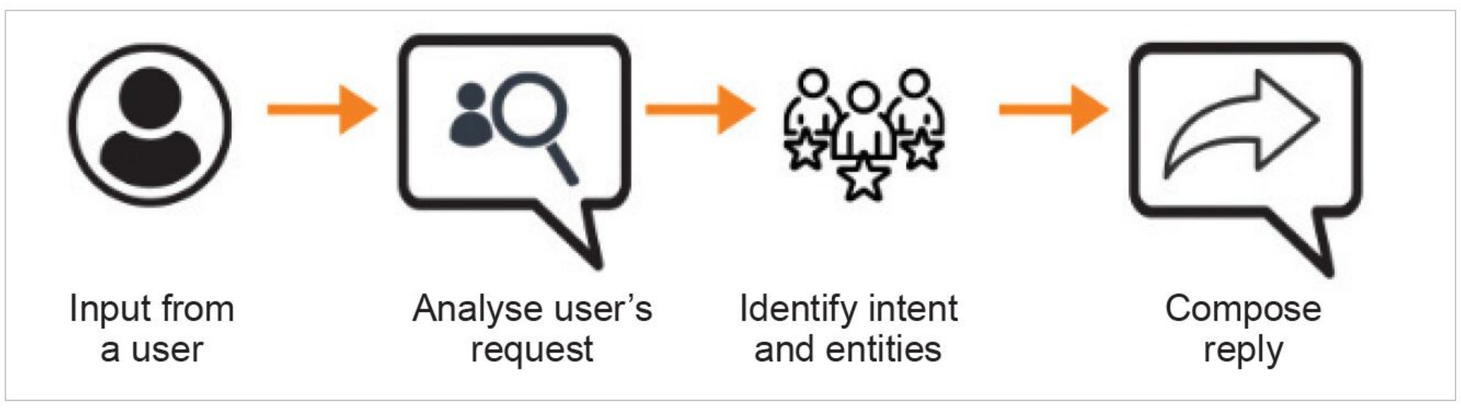


Figure 1: How AI-based chatbots work

Table 1: Popular AI-based chatbots used in industries

Company name	AI-based chatbots used for their business
Bizbike	Bizbike is Belgium's biggest e-bike provider. It has implemented AI chatbots to efficiently and reliably respond to consumer enquiries.
Starbucks	Starbucks uses AI chatbots to place orders via voice and carry out payment transactions. It also uses them for digital marketing and retail purchase.
Belfius	Belfius is a popular, trusted bank in Belgium that uses AI chatbots to manage insurance claims and respond to customer queries efficiently.
iFood	iFood is a Brazil-based company that uses an AI chatbot to manage an online meal delivery system.
European Commission	The European Commission uses an AI chatbot to manage traveller's queries.
Kotak Life Insurance	India's Kotak Life Insurance has developed the Kaya virtual assistant as an AI-based chatbot system to manage user queries.

are utilised in the travel industry for tasks like booking flights, hotels, or rental cars. They can also offer travel recommendations and information about destinations and answer frequently asked questions. Chatbots can assist in itinerary management and provide real-time updates about travel schedules. In the insurance industry, chatbots help customers file claims, obtain policy information, and answer questions about coverage. They can guide users through claims, collect relevant information, and offer instant assistance. Table 1 shows examples of how different industries are leveraging AI-based chatbots for their businesses.

Several AI-based chatbot tools are available in the market, each with its own features and capabilities (Table 2).

Using IBM Watson to develop a chatbot application

The steps we need to follow when using IBM Watson to deploy an AI-based chatbot solution are briefly listed below.

Step 1. Set up an IBM Cloud account: If you don't already have an IBM Cloud account, sign up at *https://cloud.ibm.com/registration*.

Table 2: Comparison of popular AI-based chatbots

Criteria	WP-Chatbot	Microsoft Bot	IBM Watson Assistant	CSML	LivePerson
Developed by	MobileMonkey	Microsoft	IBM	Clevy.io	LivePerson, Inc.
AI technology used	NLP	NLP	NLP, ML	NLP, ML	NLP
Key features	Facebook page branding, compatible with WordPress	Allows use of open source SDK	Provides a visual interface for designing chatbot flows, and supports multiple channels	Provides a ready-to-use chatbot templates library	Hyperlinks and custom pre-written statements
Target application	WordPress ecosystem	To build a conversational AI experience	To respond to user queries for commerce and support	To improve the customer experience for e-commerce platforms	To engage customers using messaging, voice, etc
Pricing	14-day free trial; and US$ 99/month	No free trial US$ 499/month	US$ 140/instance	Free version available	Free trial available

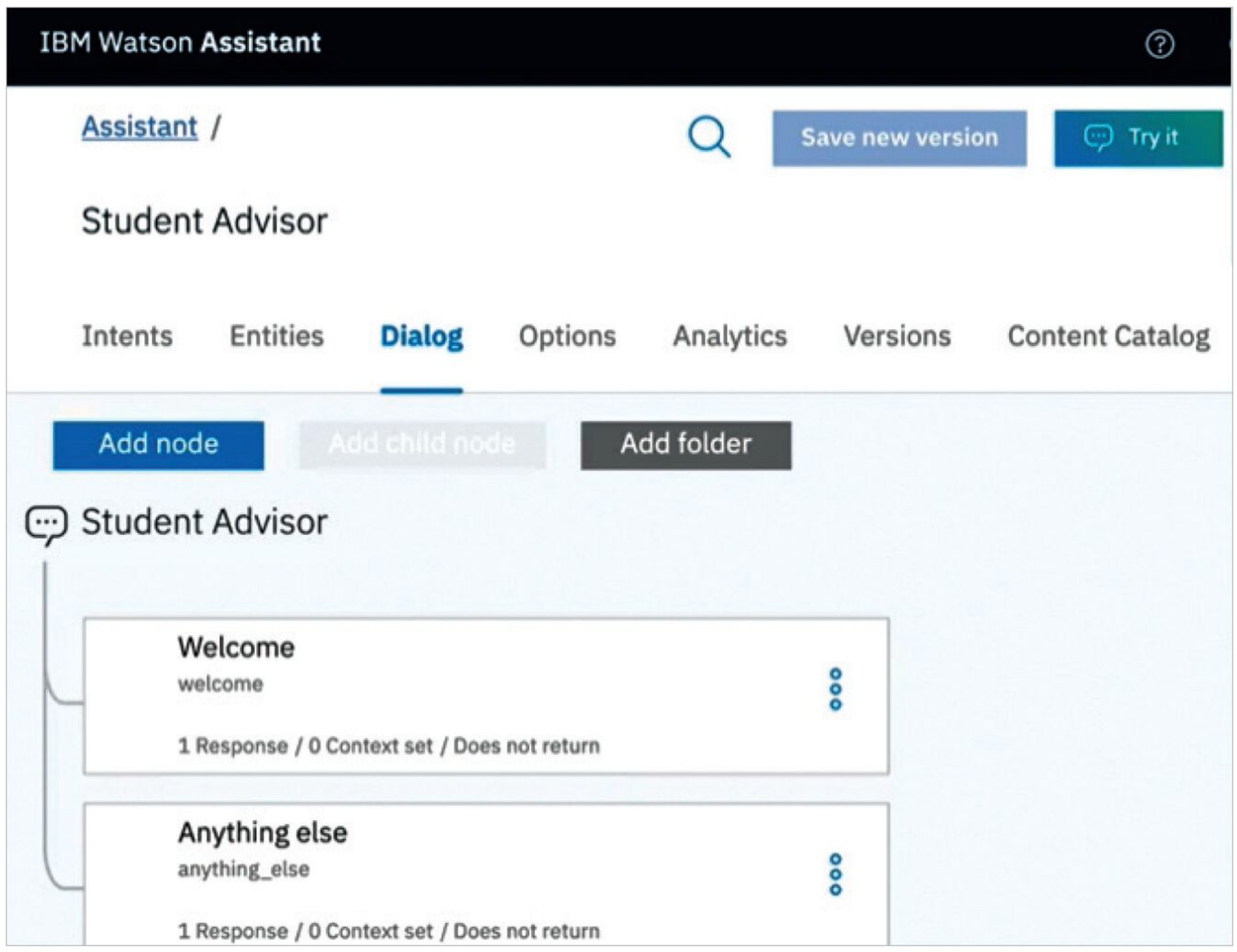


Figure 2: Create a chatbot assistant to address customer queries

Step 2. Create an IBM Watson Assistant service: Once you successfully log in at IBM Watson, create an assistant service using the catalogue menu bar. Now, generate assistant and dialogue skills. Give a name to your dialogue skill, for example, 'student advisor', as shown in Figure 2. An assistant is a chatbot that helps connect user conversation flow and search for an answer from the knowledge database.

Step 3. Design the conversation flow: Use the Watson Assistant tooling to design the conversation flow by creating nodes and defining the dialogue structure. Each node represents a step in the conversation, including user inputs and chatbot responses.

Step 4. Define intents and entities: Identify the intents (user goals or intentions) and entities (important pieces of information) that your chatbot should understand. Train Watson Assistant to recognise these intents and entities to improve the accuracy of understanding user inputs. You can construct an intent called 'feedback' that acts in response to user input, as shown in Figure 3.

Step 5. Add a custom greeting node: Click the 'Add node' button in the workspace editor to create a new node, as depicted in Figure 4. This node will represent the custom greeting.

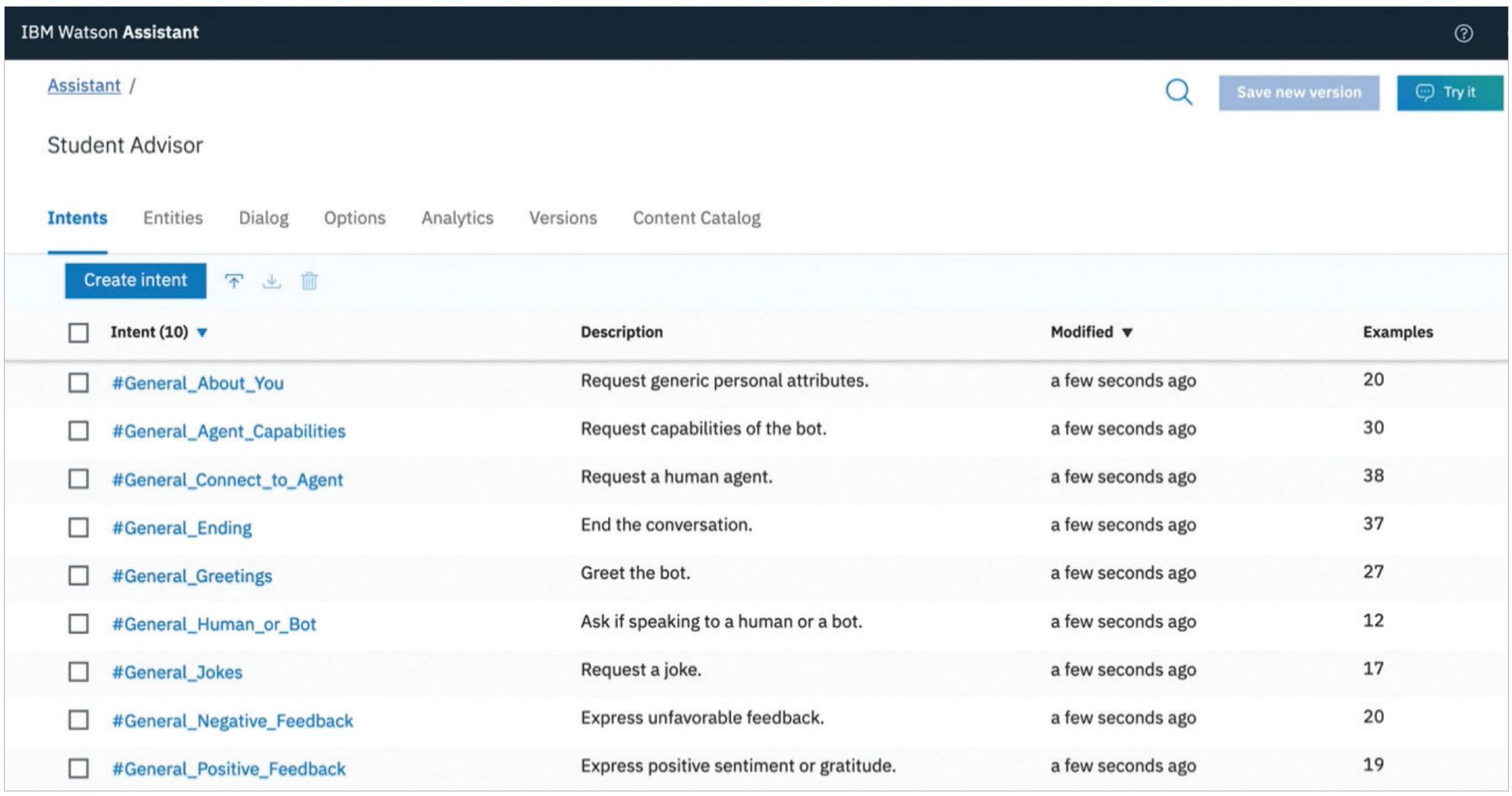


Figure 3: Create intent to define the user queries

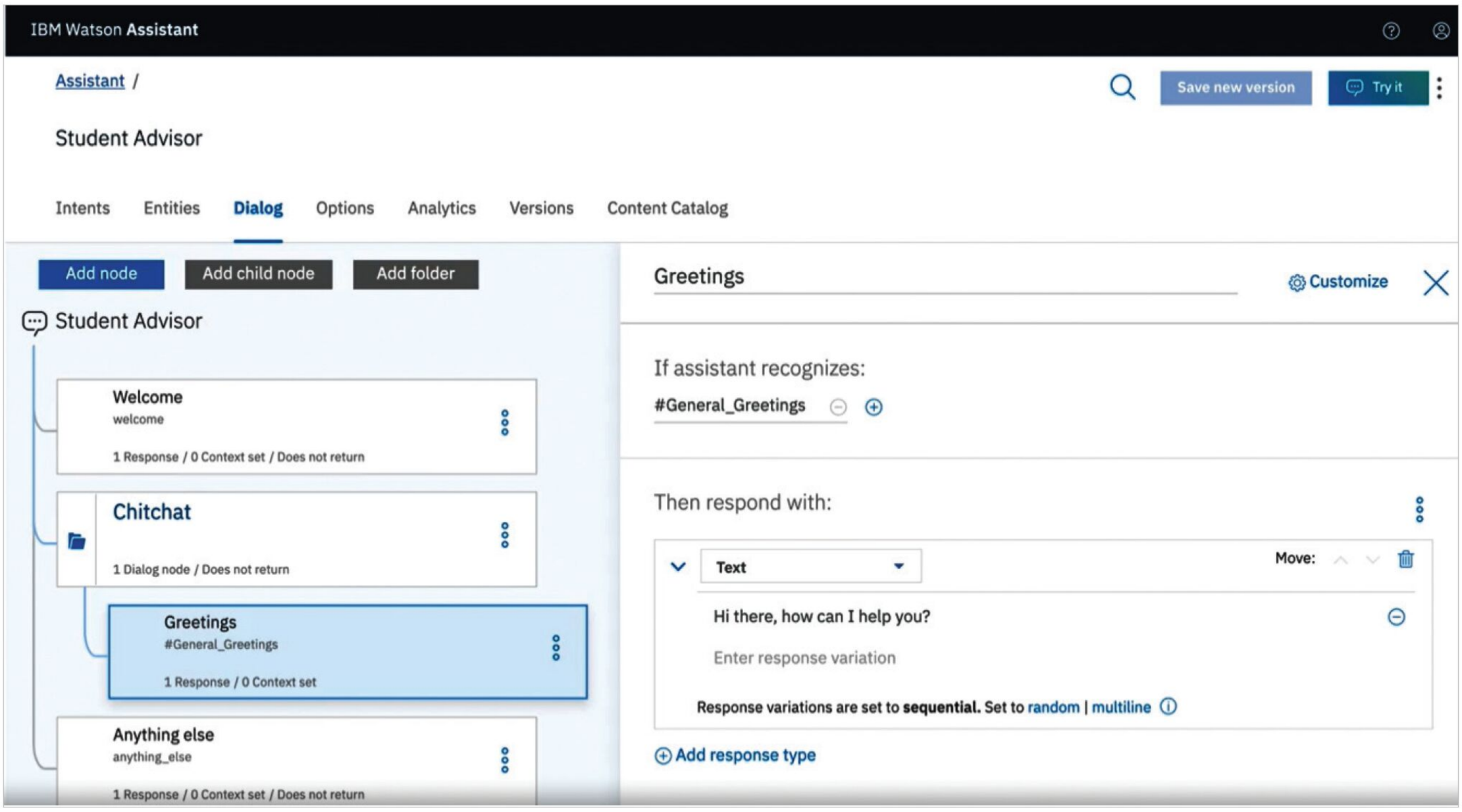
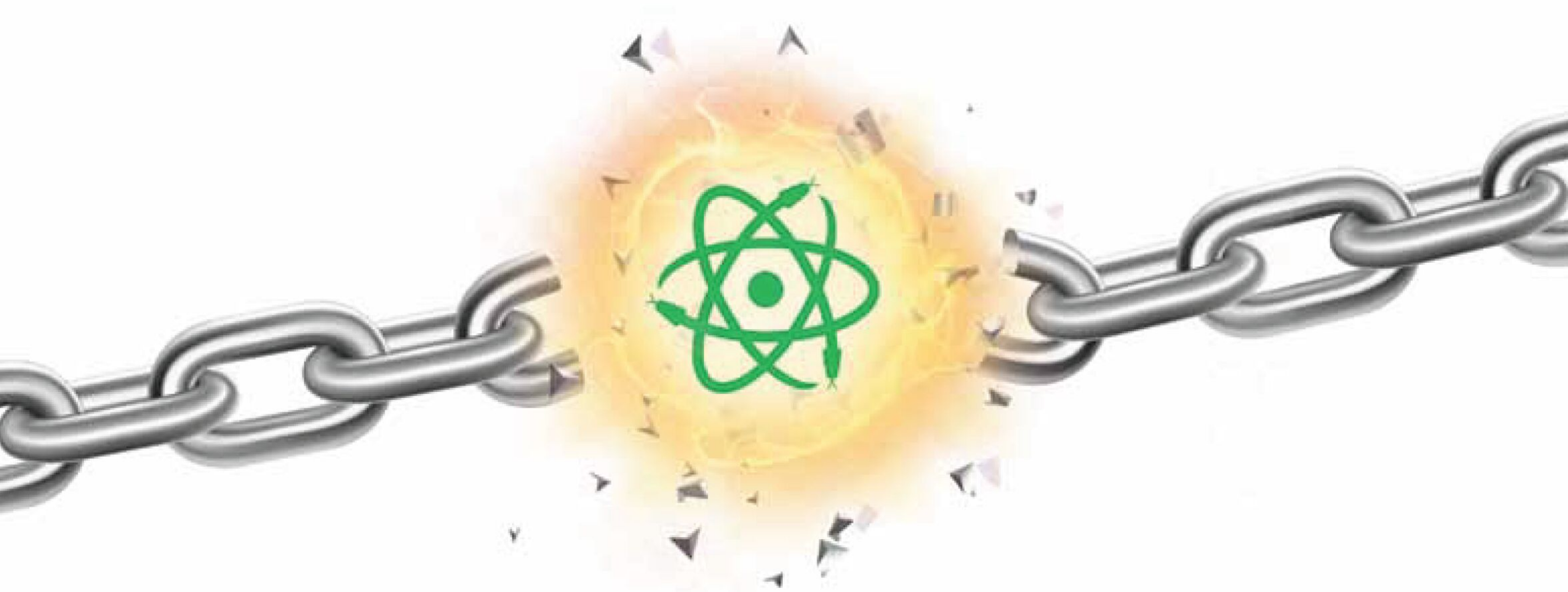


Figure 4: Adding a new custom greeting node

Continued to page...92

Unleashing the Power *of* ReactPy



This is an exploration of ReactPy, a Python library that helps create interactive user interfaces without JavaScript dependency. It's simple to use for those new to web development, and helps integrate back-end and front-end technologies.

ReactPy is a contemporary Python library that enables developers to create dynamic and interactive user interfaces, resembling the functionalities of ReactJS, while eliminating the need for JavaScript dependency (as shown in Figure 1). It offers a seamless way to create captivating front-end experiences using Python and leveraging components that resemble those found in React. The library strikes an optimal balance by providing a straightforward approach for individuals new to web development, while also offering the necessary robustness for scaling projects. Whether you are a Python enthusiast looking to improve your front-end skills or a developer seeking smooth integration between back-end and front-end technologies, ReactPy is a highly valuable asset to have in your toolkit. Let's dive in a comprehensive exploration of its capabilities, empowering you to effortlessly develop user interfaces for applications with simplicity and benefiting from the accessibility and robustness it offers.

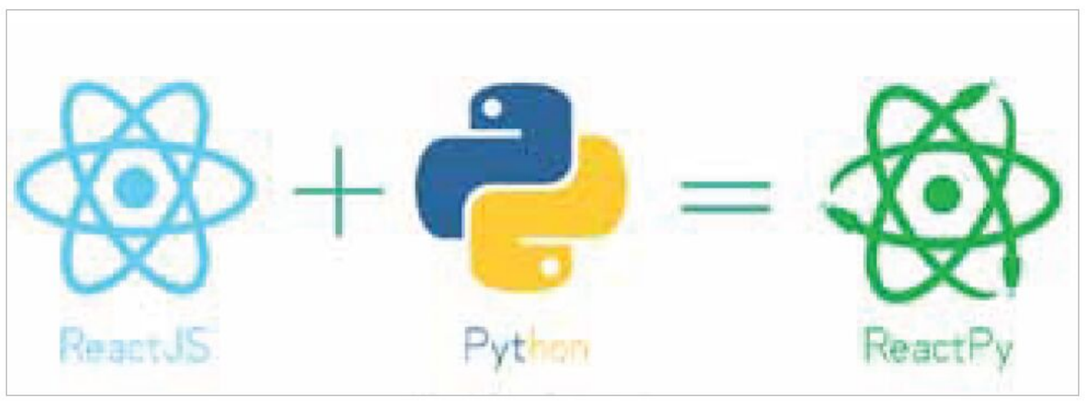


Figure 1: ReactPy

Setting up the virtual environment

First, we install *virtualenv* to create a virtual environment:

```
$ pip install virtualenv
```

After installation, we will create a virtual environment:

```
$ virtualenv -p /usr/bin/python3.10   venv
```

This will create a new directory called *venv.*

I am using Python 3.10 here; you can use Python 3.8 or 3.9. Now, activate the virtual environment as shown here:

```
ht@ext:~/sambashare/rajni$ source venv/bin/activate

(venv) ht@ext:~/sambashare/rajni$ python

Python 3.10.12 (main, Jun 7 2023, 12:45:35) [GCC 9.4.0] on
linux
Type "help", "copyright", "credits" or "license" for more
information.
```

If you want to deactivate the virtual environment, simply write the command in the terminal as:

```
(venv) ht@ext:~/sambashare/rajni/venv/code$ deactivate
```

ReactPy is typically installed using *pip* along with one of the back-ends that it comes natively supported with. You can execute ReactPy as shown in Figure 2 if you want to use Starlette as the back-end.

Let's take a simple example of ReactPy as shown below:

```
1 from reactpy import component, html, run
2 @component
3 def Show ():
4 return html.hl("Hello, world!")
5 run (Show,host="192.168.11.19",port=7000)
```

You can run this code using the Sublime editor or just the usual Python script. Save the content to *hello.py* and execute it with the following command:

```
$ python hello.py
```

Figure 3 shows how your output should appear.

Enter the web address *http://192.168.11.19:7000* into your browser. As shown in Figure 4, the 'Hello world' message is visible. If an error has occurred, change a small portion of the code, such as:

```
$ run(Show, host="0.0.0.0", port=7000)
```

…or:

```
$ run(Show)
```

If you cannot mention the host and port, then the application is running at the default IP address.

I'll break down what each line does here.

1. The first line, 'from reactpy import component,

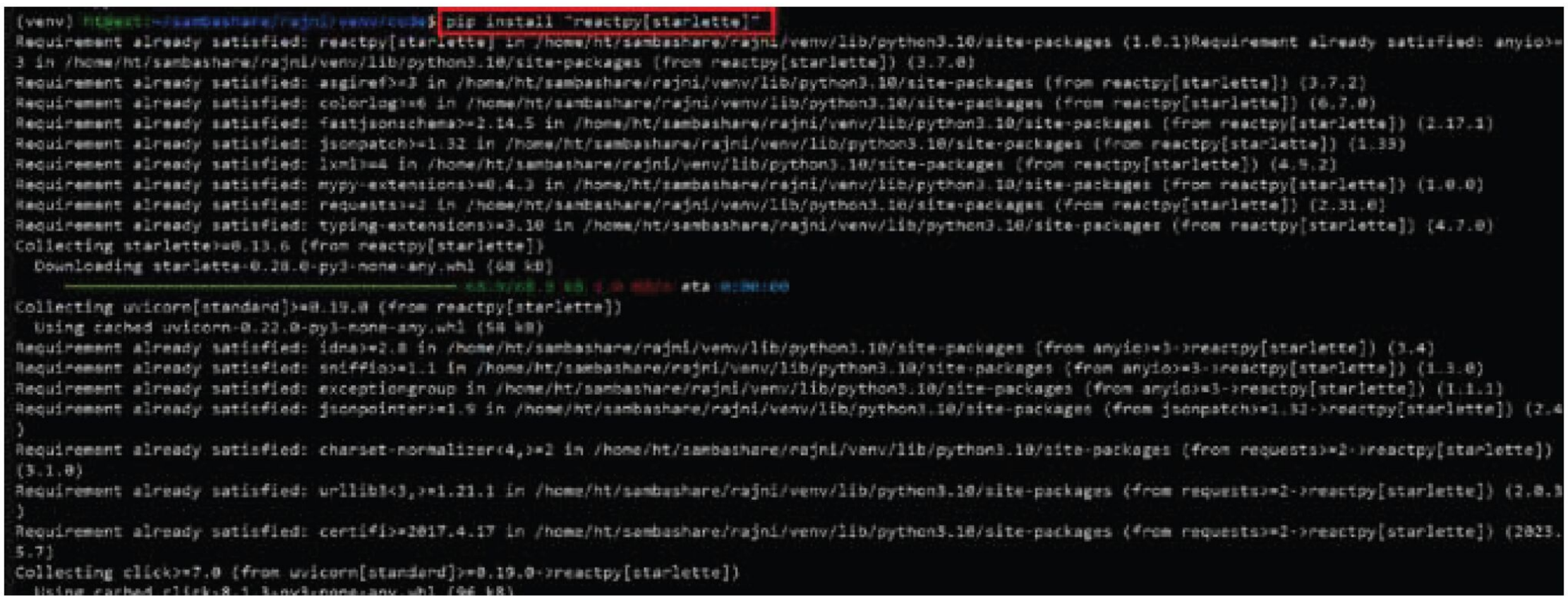


Figure 2: Installation process

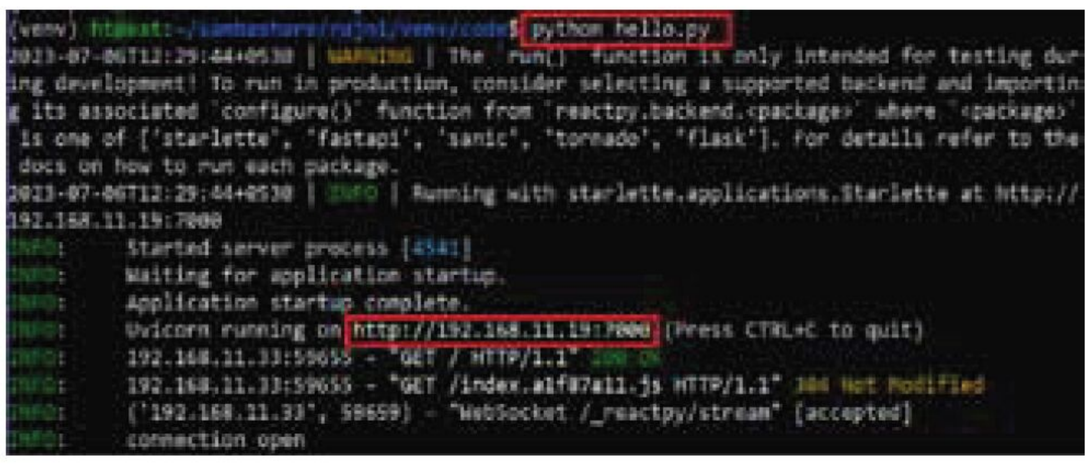


Figure 3: Output of *hello.py*

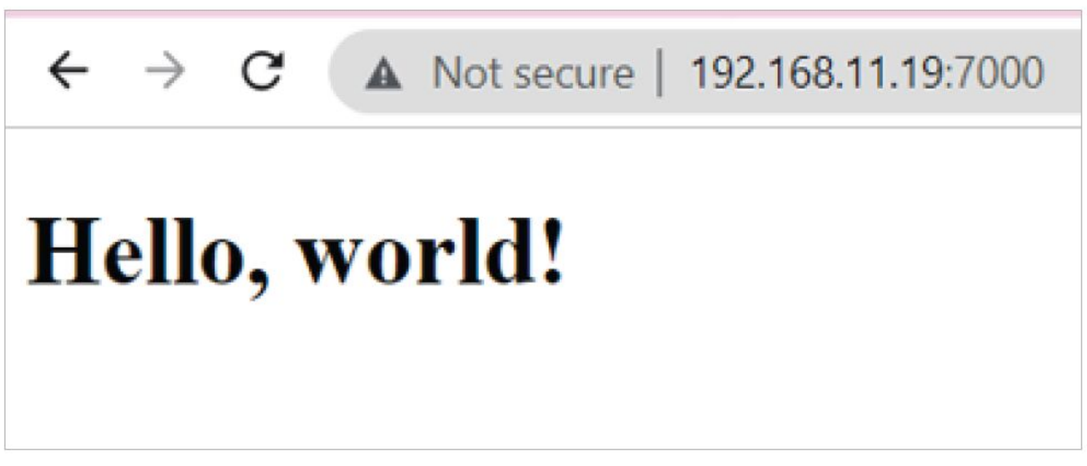


Figure 4: Output of 'Hello, world!' on the browser

html, run', imports the required components from the 'reactpy' library. 'html' offers functions to create HTML elements, 'component' is a decorator used to define React components, and 'run' is used to execute the React application.

2. The decorator '@component' designates the ensuing function as a React component.
3. 'def Show():' declares the 'Show' function, which symbolises a React component.
4. The line that reads 'return html.h1("Hello, world!")' specifies the return value for the 'Show' component. It creates an HTML heading element ('h1>') with the text 'Hello, world!' using the 'html.h1' function from the 'html' module.
5. The React application is launched with the command 'run(Show, host="192.168.11.19", port=7000)'. The application is launched using the 'run' function from the 'reactpy' library. The root component of the application is passed as the 'Show' component. The IP address and port number on which the application will be hosted are specified by the 'host' and 'port' arguments. In this instance, it will be accessible via port 7000 and IP address '192.168.11.19'.

The code above creates a straightforward React component called 'Show' that renders an HTML heading element with the text 'Hello, world!' After that, the React application is launched and hosted at the given IP address and port.

Creating a 'To-do List' application

We can create a 'To-do List' application with the help of ReactPy as follows:

```
from reactpy import component, html, run
@component
def DataList(items, filter_by_priority=None, sort_by_
priority=False):
    if filter_by_priority is not None:
        items = [i for i in items if i["priority"] <= filter_
by_priority]
    if sort_by_priority:
        items = sorted(items, key=lambda i: i["priority"])
    list_item_elements = [html.li({"key": i["id"]},
i["text"]) for i in items]
    return html.ul(list_item_elements)
@component
def TodoList():
tasks = [
        {"id": 0, "text": "Read a book", "priority": 0},
        {"id": 1, "text": "Meet Geet", "priority": 0},
        {"id": 2, "text": "Hit the gym", "priority": 2},
        {"id": 3, "text": "Buy egg", "priority": 1},
        {"id": 4, "text": "Clean the house", "priority": 2},
        {"id": 5, "text": "organize office", "priority": 2},
        {"id": 6, "text": "Do some coding", "priority": 1},
        {"id": 7, "text": "Read a book", "priority": 1},
    ]
    return html.section(
        html.h1("My Todo List"),
        DataList(tasks, filter_by_priority=1, sort_by_
priority=True),
    )
run(TodoList, host="192.168.11.19",port=7000)
```

The output of the To-do List application is shown in Figure 5.

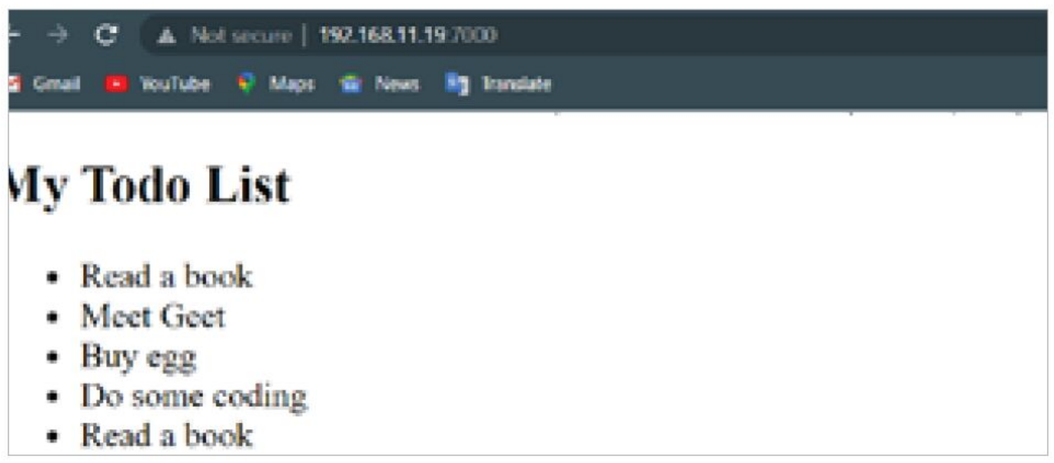


Figure 5: Output of the 'To-do List' application on the browser

Let's move on to something more challenging. You can add images, titles, and captions, and combine them into one function by using various HTML elements:

```
from reactpy import component, html, run
@component
def Title(title):
    return html.h1(title)
@component
def Photo():
    return html.img(
        {
                "src":"https://cdn.media.
amplience.net/i/epammarketplace/open_source_license_
header?version=1684831857337",
            "style": {"width": "40%"},
        }
    )
  @component
def PhotographerName(caption):
    return html.h4(caption)
  @component
def PhotoViewer():
    return html.section(
        Title("Picture of the day"),
        Photo(),
        PhotographerName("Visos Banner")
    )
run(PhotoViewer, host="192.168.11.19",port=7000)
```

The result is displayed in Figure 6.



Figure 6: Final output

You can see that your code is reusable. So, for example, you can define a new function that has the photo and photographer name, and after that use that function multiple times, specifying only the URL and name itself. This is enough for making something like a gallery website.

ReactPy back-end

ReactPy supports the following back-end implementations:

- FastAPI
- Flask
- Sanic
- Starlette
- Tornado

To install ReactPy back-end, run the following command; you can install more than one option by separating them with commas.

```
$ pip install "reactpy[fastapi,flask,sanic,starlette,tornado]"
```

As can be seen, with ReactPy, building dynamic user interfaces is a breeze. END

By: Rajnesh Devi

The author is assistant professor at the Yamuna Group of Institutions.

Continued from page...88

Step 6. Save and test: Finally, save your changes and try the custom greeting node by interacting with the chatbot to ensure it triggers the greeting when the specified input is provided. A sample deployed application is shown in Figure 5.

Chatbots are gaining popularity in industries with their ability to enhance customer experiences, improve operational efficiency, reduce costs, and provide businesses with valuable insights for better decision-making. IBM Watson is a powerful platform that offers a range of AI capabilities, including natural language processing (NLP) and machine learning, which can be leveraged to build AI chatbots. Businesses can build robust and intelligent chatbot solutions by utilising IBM Watson's AI capabilities. END

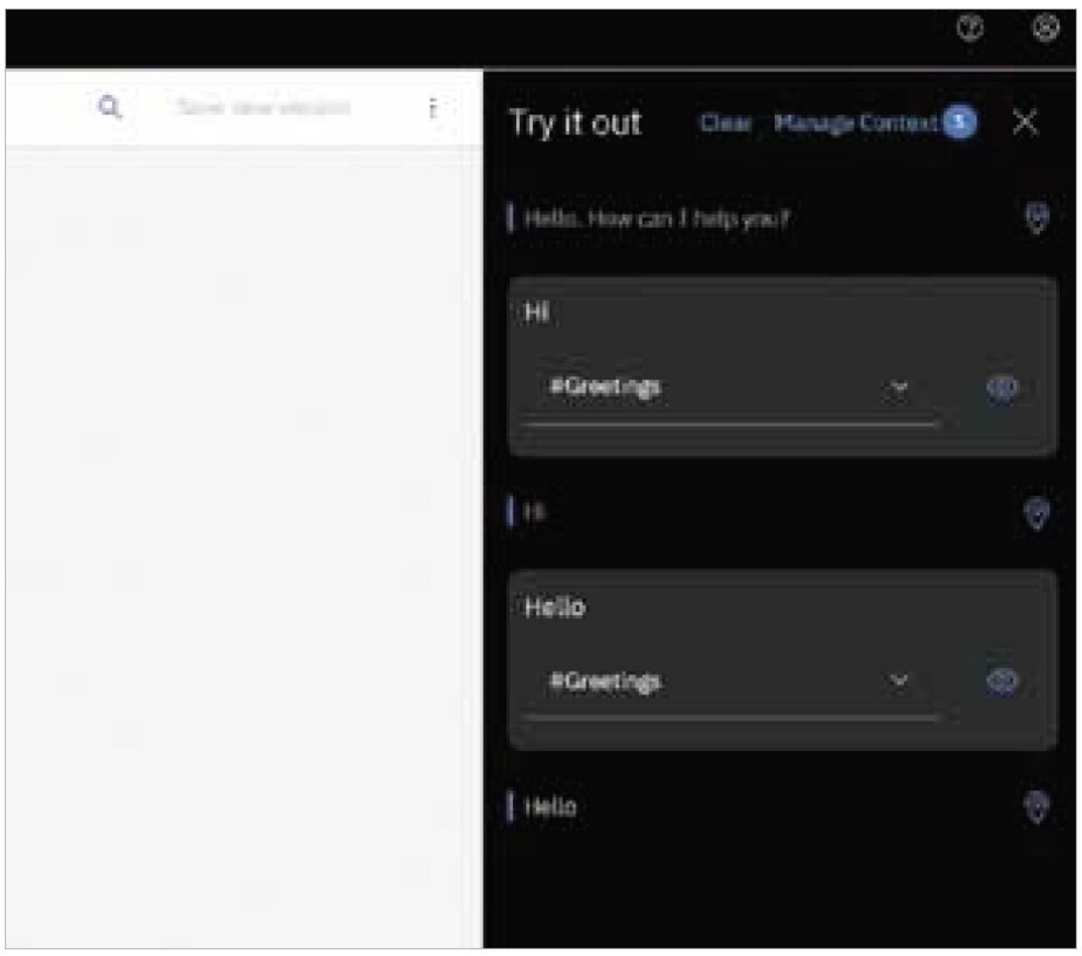


Figure 5: Test the greetings chatbot you developed

By: Dr Aditya Bhardwaj

The author is a B. Tech, 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 is experienced in cloud computing and open source technology.

References

- *https://www.ibm.com/in-en/watson*
- *https://www.ibm.com/topics/chatbots*

Mastering Immutable Infrastructure:
A Step-by-Step Guide to Building Golden AMI

You can leverage Packer, Ansible, and Testinfra to automate the creation and testing of golden AMIs, ensuring your infrastructure is configured correctly. Here's how!

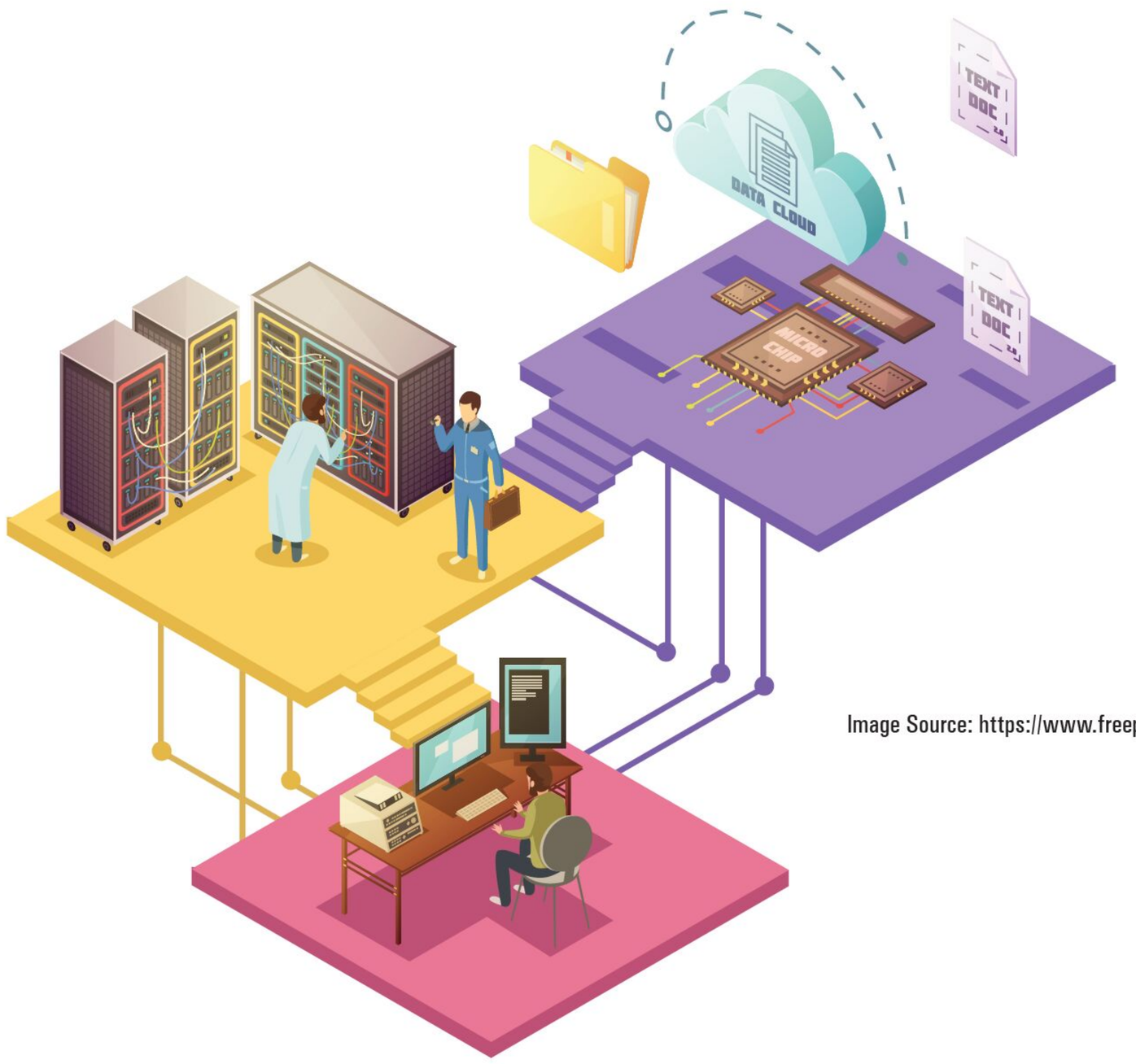


Image Source: https://www.freepik.com

This article covers a step-by-step process of building a golden AMI using Packer, Ansible and TestInfra. It also demonstrates how to integrate these tools to master immutable infrastructure.

In recent years, immutable infrastructure has become the most popular approach to deploying applications in the cloud. It enables developers to create secure, scalable, and repeatable infrastructure. Immutable infrastructure involves creating machine images, also known as golden AMIs, that are pre-configured and ready to run applications. These machine images are then deployed as-is and never modified, ensuring reliable and consistent infrastructure.

Before we begin, let's quickly understand the basic terminologies and get familiar with the tools we are going to use.

- ***Immutable infrastructure****:* This approach to infrastructure management emphasises creating and deploying infrastructure in a non-changing state. In other words, an immutable infrastructure cannot be modified once created. Instead of modifying the existing infrastructure, a new version is created every time changes are made. Immutable infrastructure has several benefits, such as easier testing, more efficient deployments, and better security.
- ***Golden AMI****:* It is an Amazon machine image containing the operating system, configuration, and applications required to run an application or service. It is the foundation for creating instances in AWS. Ensuring that the AMI is up-to-date, secure, and consistent is crucial.
- ***Packer****:* This tool creates identical machine images for multiple platforms from a single source configuration. It allows you to automate the process of building machine images, which can be used for deployment, testing, and development. Packer supports multiple builders, including Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure.

- ***Ansible:*** This open source automation tool enables you to automate applications and systems' deployment, configuration, and orchestration. It is designed to be simple, lightweight, and extensible.
- ***Testinfra:*** It is a framework that enables you to write unit tests for infrastructure. It provides a simple syntax for defining test cases, which can be used to verify the functionality and configuration of infrastructure components.

Now that we have a basic understanding of the immutable infrastructure concepts, let's dive in and understand how to deploy Apache web server on an Amazon Linux 2 machine by building a flawless golden AMI using Packer, configuring and deploying the web server with Ansible, and testing its functionality with Testinfra. Before we begin, make sure you have the following:

- Access to AWS console to launch EC2 instance.
- Packer, Ansible, and Testinfra packages on the local machine.
- Basic understanding of Ansible, Packer and Python.

Steps to building a golden AMI

Define Packer template: First, we'll create a Packer template that defines the AMI we want to bake in. Create a 'webserver.pk.hcl' file that installs 'httpd' package, configures a virtual host 'helloworld.example.com', and runs the web server on an Amazon Linux 2 AMI.

```
variable "app_role" { type = string }
variable "app_version" { type = string }
variable "instance_type" { type = string }
variable "root_volume_size" { type = string }
variable "aws_region" { type = string }
variable "aws_access_key" { default = "${env("aws_access_
key")}" }
variable "aws_secret_key" { default = "${env("aws_secret_
key")}" }

locals {
    timestamp = regex_replace(timestamp(), "[- TZ:]", "")
    ami_name  = lower("${var.app_role}-${var.app_version}")
}

source "amazon-ebs" "webserver" {
  region                  = "${var.aws_region}"
  ami_name                = "${local.ami_name}-${local.
timestamp}"
  ami_description         = "Golden AMI To Deploy WebServer"
  ssh_username            = "ec2-user"
  ssh_wait_timeout        = "10000s"
  instance_type           = "${var.instance_type}"
  source_ami              = "ami-052f483c20fa1351a"
  launch_block_device_mappings {
    device_name           = "/dev/sda1"
    volume_size           = "${var.root_volume_size}"
    volume_type           = "gp3"
    delete_on_termination = true
  }

  tags = {
    Name = "${var.app_role}"
  }
}

build {

  sources = ["source.amazon-ebs.webserver"]

  provisioner "file" {
    source          = "test_apache.py"
    destination     = "/tmp/"
  }

  provisioner "ansible" {
    playbook_file = "webserver.yaml"
  }

  provisioner "shell" {
    execute_command  = "{{.Vars}} sudo -E -S bash '{{.
Path}}'"
    inline = [
      "python3 -m pytest /tmp/test_apache.py",
    ]
  }
}
```

Define Ansible playbook: In this step, we will define an Ansible playbook that installs 'http' package, 'pytest-testinfra' module, configures virtual hosts, and finally starts the 'httpd' service on the Amazon Linux2 machine during the baking process.

```
- name: WebApp Playbook
  hosts: all
  become: true

  tasks:
    - name: install_httpd
      yum:
        name: httpd
        state: present

    - name: set_listener
      lineinfile:
```

```
    path: /etc/httpd/conf/httpd.conf
    regexp: '^Listen\s+80$'
    line: 'Listen 0.0.0.0:80'

- name: create_vhost
  copy:
    content: |
      <VirtualHost *:80>
        ServerName helloworld.myexample.com
        DocumentRoot /var/www/html
        ErrorLog /var/log/httpd/error.log
        CustomLog /var/log/httpd/access.log combined
        <Directory /var/www/html>
            AllowOverride All
            Require all granted
        </Directory>
      </VirtualHost>
    dest: /etc/httpd/conf.d/helloworld.conf

- name: create_webroot
  file:
    path: /var/www/html
    state: directory
    owner: apache
    group: apache
    mode: '0755'

- name: create_index_html
  copy:
    content: '<h1>Hello, World!</h1>'
    dest: /var/www/html/index.html
    owner: apache
    group: apache
    mode: '0644'

- name: start_httpd
  service:
    name: httpd
    state: started
    enabled: true

- name: install_testinfra
  ansible.builtin.pip:
    name: pytest-testinfra
    executable: pip3
```

Define the Testinfra test cases: Once the templating part is done, we will need to write a simple test case using Testinfra to ensure the AMI has 'httpd' package successfully installed, configured, and running fine. In short, the AMI should be in the desired state. Here is an example of a sample test case for 'httpd' service:

```
import testinfra

def test_apache_is_installed(host):
    assert host.package("httpd").is_installed

def test_apache_is_running(host):
    assert host.service("httpd").is_running
    assert host.service("httpd").is_enabled

def test_apache_virtual_host(host):
    virtual_host = """
    <VirtualHost *:80>
        ServerName helloworld.myexample.com
        DocumentRoot /var/www/helloworld
        ErrorLog /var/log/httpd/helloworld-error.log
        CustomLog /var/log/httpd/helloworld-access.log
combined
        <Directory /var/www/helloworld>
            AllowOverride All
            Require all granted
        </Directory>
    </VirtualHost>
    """
    assert host.file("/etc/httpd/conf.d/helloworld.conf").
exists
    assert host.file("/etc/httpd/conf.d/helloworld.conf").
contains(virtual_host)
def test_apache_index_html(host):
    assert host.file("/var/www/html/index.html").exists
    assert host.file("/var/www/html/index.html").
contains("<h1>Hello, World!</h1>")
```

Bake the golden AMI: Now, we can start baking the AMI once we are done with all the steps mentioned above. So let's run the following command, which will initiate the baking process:

```
$ packer build -var-file=templates/webserver.pkrvars.hcl
templates/webserver.pkr.hcl

amazon-ebs.webserver: output will be in this color.

==> amazon-ebs.webserver: Prevalidating any provided VPC
information
==> amazon-ebs.webserver: Prevalidating AMI Name:
webserver-2.4-20230520103749
    amazon-ebs.webserver: Found Image ID: ami-
028a473e64001bfb2
==> amazon-ebs.webserver: Provisioning with Ansible...
    amazon-ebs.webserver: Setting up proxy adapter for
Ansible....
****************************************************
```

```
    amazon-ebs.webserver: changed: [default]
    amazon-ebs.webserver:
    amazon-ebs.webserver: PLAY RECAP ************************
amazon-ebs.webserver: default                    :
ok=8    changed=7    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
    amazon-ebs.webserver:
==> amazon-ebs.webserver: Provisioning with shell script: /
tmp/packer-shell1563525277
    amazon-ebs.webserver: ============================= test
session starts ==============================
    amazon-ebs.webserver: platform linux -- Python 3.7.16,
pytest-7.3.1, pluggy-1.0.0
    amazon-ebs.webserver: rootdir: /tmp
    amazon-ebs.webserver: plugins: testinfra-7.0.0
    amazon-ebs.webserver: collected 4 items
    amazon-ebs.webserver:
    amazon-ebs.webserver: ../../tmp/test_apache.py ...
.                                                [100%]
    amazon-ebs.webserver:
    amazon-ebs.webserver: ============================== 4
passed in 0.20s ===============================
==> amazon-ebs.webserver: Stopping the source instance...
    amazon-ebs.webserver: Stopping instance
==> amazon-ebs.webserver: Waiting for the instance to stop...
==> amazon-ebs.webserver: Creating AMI
webserver-2.4-20230520103749 from instance
i-000ad46d12682b5c2
    amazon-ebs.webserver: AMI: ami-0a78efa7e31b4d102
```

Deploy the golden AMI: The last step is to deploy the AMI in pre-production or production upon a successful run. Infrastructure changes can be deployed through CI/CD automation tooling like Jenkins, CircleCI, CloudFormation, or Terraform. Such automation ensures the infrastructure is consistent and easily replicated across multiple environments.

Building golden AMIs is an important step in mastering immutable infrastructure. By following the steps outlined in this guide, you can use Packer, Ansible, and TestInfra to automate the process of building and testing your AMIs and ensure that your infrastructure is always configured correctly. By doing so, you can increase the reliability and security of your infrastructure and reduce the risk of downtime or security breaches. END

Reference

- *https://github.com/vijayg92/ec2-golden-ami-demo*
- *https://developer.hashicorp.com/packer/docs*
- *https://testinfra.readthedocs.io/en/latest*
- *https://docs.ansible.com*

By: Vijay Singh Gosai

The author is an open source enthusiast and currently works as a senior site reliability engineer. His interest areas include distributed systems, cloud computing, infrastructure automation, software development, design patterns, DevOps, and emerging technologies.

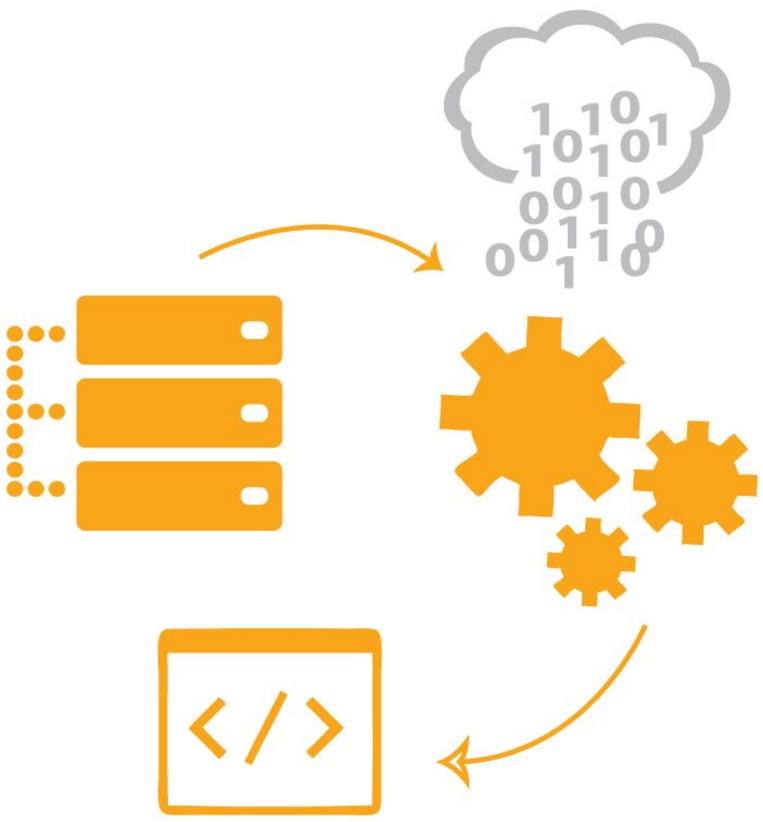
OSFY Magazine Attractions During 2023-24

Month	Theme
March 2023	Security, Network Management and Monitoring
April 2023	Open Source Programming (Languages and tools)
May 2023	Cloud Special: Everything from management to implementation
June 2023	AI, Deep learning and Machine Learning
July 2023	Database management and Optimisation
August 2023	Mobile/Web App Development, Optimisation and Security
September 2023	DevOps Special
October 2023	Blockchain and Open Source
November 2023	Open Source and IoT and Edge
December 2023	All About Data Management
January 2024	Containers and Managing Containers
February 2024	Open Source on Windows and Best in the world of Open Source (Tools and Services)

Cloud Orchestration with Infrastructure as Code (IaC): The Benefits

IaC is the art of defining infrastructure using code, making it repeatable and predictable. With cloud orchestration, you can deploy and manage your infrastructure in a more agile and efficient way. No more manual configurations or guesswork!

What is it that makes Infrastructure as Code (IaC) one of the greatest technologies in the cloud scenario of the 21st century? Well, it is IaC that powers cloud infrastructure. Supported by various highly innovative features and tools, IaC automates IT infrastructure, empowering organisations to quickly build, deploy, and scale cost-efficient cloud applications.



IaC: A blessing for cloud optimisation?

Infrastructure encompasses various components required to deploy software in a cloud environment, such as VMs, containers, serverless functions, security features like IAM (identity and access management) and KMS (key management service), networking, and monitoring and logging capabilities.

Now, let's look at the Google Play Store. You download tens to hundreds of apps to your Android device from there. In the past, updating these apps required manual effort. You had to individually check for updates and install them. However, with IaC in place, the process becomes seamless and automated. Through well-defined code and configuration files, the infrastructure behind the Play Store can automatically detect app updates and efficiently roll out the new versions to users' devices. This ensures that users have the latest features and bug fixes without the need for manual intervention. Thanks to IaC, your apps are now automatically updated.

Virtualisation and cloud development eliminate the challenges of physical hardware management, allowing developers to build their own virtual servers or containers.

As technology rapidly evolves, it is important for IT infrastructure development staff to set up and update the cloud environment from time to time. Data administrators need to determine the resources required for coding, ranging from network configuration to data storage in cloud environments.

IaC: A force to be reckoned with in cloud orchestration

IaC is playing an important role in IT supporting frameworks like servers, hard disk drives, etc; applying security updates, installing services and applications, establishing networks, and managing memory. It overcomes common IT problems such as configuration drifting, operator mistakes, inefficiencies and information loss. Flexible licensing models and reduced maintenance costs are a few factors contributing to the dominance of IaC in cloud orchestration.

With evolving consumer behaviour and demands, enterprises must act to stay competitive in the market. Consequently, cloud infrastructures are expanding and IaC is the only method that can handle the complexities associated with them.

Large-scale organisations, such as those in telecommunications, retail, and manufacturing sectors, often struggle to meet customer demands due to work delays. To increase work speed and minimise human errors, automation of cloud infrastructure becomes essential for such organisations. Hence, they are increasingly opting for IaC.

The IaC market size is projected to grow from US$ 0.8 billion in 2022 to US$ 2.3 billion in 2027, at a CAGR of 24% during the forecast period.

Strategies for joining the big league of cloud orchestration

Cloud orchestration has experienced various waves of development. In the first wave, infrastructures were fairly static. The second wave introduced more complexity with the use of containers, Docker and Datadog. Modern cloud infrastructure is even more complex, leveraging serverless and managed services as part of cloud applications.

It is important to note that all major cloud providers have their own version of IaC, using different languages and syntax to render code. Table 1 provides an overview.

Developers have gained enormous potential and convenience from the cloud with its elastic capabilities and API based architectures. However, GitHub and Bitbucket have also become valuable sources for attackers. These attackers leverage automation to quickly and efficiently harvest stray credentials, creating more opportunities.

In the current environment, automation is omnipresent, particularly in the form of ALOps, robotic process automation (RPA), and other machine-driven tools. Consequently, a multi-cloud environment needs higher levels of automation.

Configuration tracking and capabilities are becoming essential in the modern cloud context, especially in sectors like healthcare. Cloud administrators often need to detect idle users, disconnect them, drain hosts, and shut down unnecessary servers to minimise runup costs.

They need to monitor service accounts, cloud billing accounts and side effects of automation such as large scale cloud costs.

IaC tools for cloud orchestration

IaC tools include AWS CloudFormation, Red Hat Ansible, Chef, Puppet, SaltStack and HashiCorp Terraform. Some tools rely on domain-specific languages (DSLs), while others use standard template formats like YAML and JSON.

Cloud Provider	IaC	Language used
Google Cloud Platform	Deployment Manager	YAML
Amazon Web Services (AWS)	Cloud Formation	YAML or JSON
Microsoft Azure	Azure Resource Manager	JSON

Table 1: The different versions of IaC developed and used by the major cloud providers

Terraform: Terraform is a declarative provisioning and infrastructure orchestration tool that allows engineers to automate all aspects of their cloud-based infrastructure.

It works with all the leading cloud providers and enables the automation of resource deployment across multiple providers in parallel, regardless of the physical location of servers, DNS servers, or databases. It can handle applications written in any programming language.

SCARLETEEL, a recent cyber attacker, is an excellent example of how IaC machinery was exploited to expand its reach. In this case, Terraform was used by the cloud administration, and files were stored in an S3 bucket.

Another way to define IaC is by using a well-known programming language. For instance, Pulumi can be used to write IaC using TypeScript, JavaScript, Python, Go, and .NET.

Red Hat's Ansible: Red Hat's Ansible is an open source automation platform that enables IT automation across cloud, on-premises, and virtualised and containerised environments. It reduces human capital, and develops more repeatable and standardised processes. Its use of the YAML language makes it more accessible. The open source model of Ansible allows anyone to contribute ideas and share them across the Ansible user community.

IBM, which acquired Red Hat for US$ 34 billion in 2019, has integrated Ansible into its Global Technology Services (GTS) organisation, a division of IBM Services.

SaltStack: SaltStack, also known as Salt, is a configuration management and orchestration tool. It uses a central repository to provide new servers and make changes to existing ones. It also helps with software installation in various IT environments, including physical and virtual servers, as well as the cloud.

Chef: Terraform is a provisioning tool, while Chef is a configuration management tool. Terraform provisions the servers, while Chef manages the software already installed on the servers.

Puppet: Puppet allows the definition of required software and configuration for a system, maintaining a specified state after the initial setup. It uses a declarative domain specific language (DSL) that is similar to Ruby to define configuration parameters for specific environments or infrastructures.

The future of IaC depends upon technological advancements and the attitudes of cloud administrators. The objective of cloud automation is not to replace humans with machines but to make work life more meaningful. END



References

- 'What is IaC? (Tutorial)' by Beau Carnes
- 'Introduction to IaC in Multi-Cloud Environment'; *onpointcorp.com/*
- IaC market information by *marketsandmarkets.com*
- 'What is Infrastructure as Code (IaC)?' as published in *ibm.com*
- Certain articles related to IaC carried in *computerweekly.com*

By: Vinayak R. Adkoli

The author holds a B.E. in industrial production, and has served as a lecturer in three different polytechnics for 10 years. He is also a freelance writer and cartoonist.

CALLING EXPERTS

Share your knowledge at the 20th edition of
India's most exciting convention on Open Source!

20th Edition

OPEN SOURCE INDIA

19-20 OCT 2023

NIMHANS CONVENTION CENTER, BENGALURU

The Who's Who of Open Source Speak Here...

Open Source India (OSI) is the premier Open Source conference in Asia targeted at nurturing and promoting the Open Source ecosystem in the subcontinent. Started as LinuxAsia in 2004, OSI has been at the helm of bringing together the Open Source industry and the community.

Scan to Register

OR visit:
www.opensourceindia.in

www.OpenSourceIndia.in

R N I No. DELENG/2012/49440, Mailed on 27/28th of Advance month
Published on 27th August 2023
Delhi Postal Regd. No. DL(S)-01/3443/2022-24
Licenced to Post without Pre-Payment Licence No U(S)-56/2022-24
Posted at LPC Delhi RMS Delhi 110006

India's No.1 Electronics Magazine
RANKING AS PER ELCINA-IMRB REPORT

"I started reading it when I was a student...
...and I am still reading it, as a student"

—CEO, Design House



electronics FOR YOU
YOURS SINCE 1969

To Subscribe:
https://subscribe.efy.in

OR

Scan This Code

For any query, call: +91-98111-55335 or email: support@efy.in