Should You Switch To Serverless Computing?

Are You Ensuring Open Source Compliance? If Not, It's Time To Do So

₹120
ISSN-2456-4885

OpenSource ForYou

Volume: 10 | Issue: 06 | Pages: 108 | April 2022

THE COMPLETE MAGAZINE ON OPEN SOURCE

EFY GROUP Publication

Enhance Your Programming Skills



Shell Scripting Is Still Going Strong

Python 3: What's New?

DAML: The Programming Language For Smart Contracts In A Blockchain



Interview With **Pradeep Sharma,** Senior Developer Relations Expert, Jina AI

A Quick Look At Cloud Cost Management Tools

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.

Does Your Antivirus Solution Provide You With Complete Protection?

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



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

- Ransomware File Protection
- ATP- Advance Threat Protection
- EDR- Endpoint Detection And Response
- Application Blocking
- Deep Learning Malware Analysis
- Exploit Prevention
- URL Blocking
- Disk and Boot Record Protection
- Peripheral Control
- Respond Investigate Remove (Root Cause Analysis)

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

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

ITS
A Total Solutions Company

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

- Mumbai
- Jaipur
- Chandigarh

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

CONTENTS

APRIL 2022 | ISSN-2456-4885

FOR U & ME

19 Are You Ensuring Open Source Compliance? If Not, It's Time to Do So

FOCUS

47 Python 3: What's New?

53 Shell Scripting is Still Going Strong

57 Plotting Data in R: Graphs

DEVELOPERS

63 SQLx: The Rust SQL Toolkit

67 What is Federated Learning?

72 Simplifying Cloud Native Development with Skaffold

75 Supervised Machine Learning with KNIME

79 Automatic Certificate Generation Using Python

ADMIN

82 Integrating Zeek with ELK Stack

89 WiFi 6 Promises Much More than Faster Speeds

92 SSL Certificates: Make the Right Choice

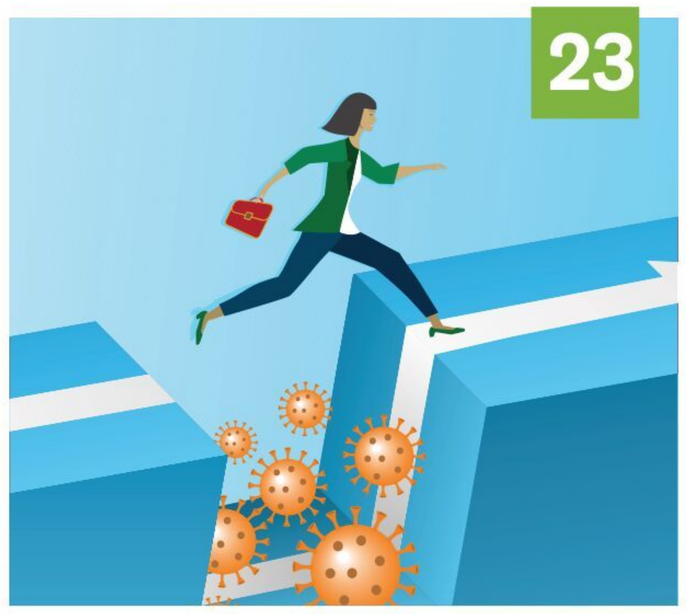
OPEN GURUS

100 Build a Smart Parking System for a Metro Station

104 Why Edge Computing is Important for IoT Applications



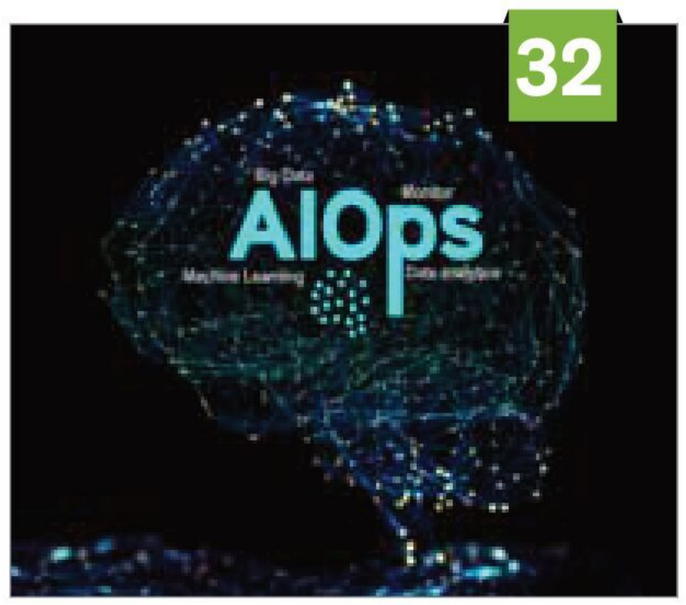
"The future is SaaS plus open source" — Pradeep Sharma, senior developer relations expert, Jina AI



Pandemic Pains: Building A Startup in the Post-Covid World



Greenfield Technology: How Open Source Software can be Useful



Why AIOps is Essential for Businesses Today



REGULAR FEATURES

07 FossBytes

www.opensourceforu.com

Job Guaranteed.

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

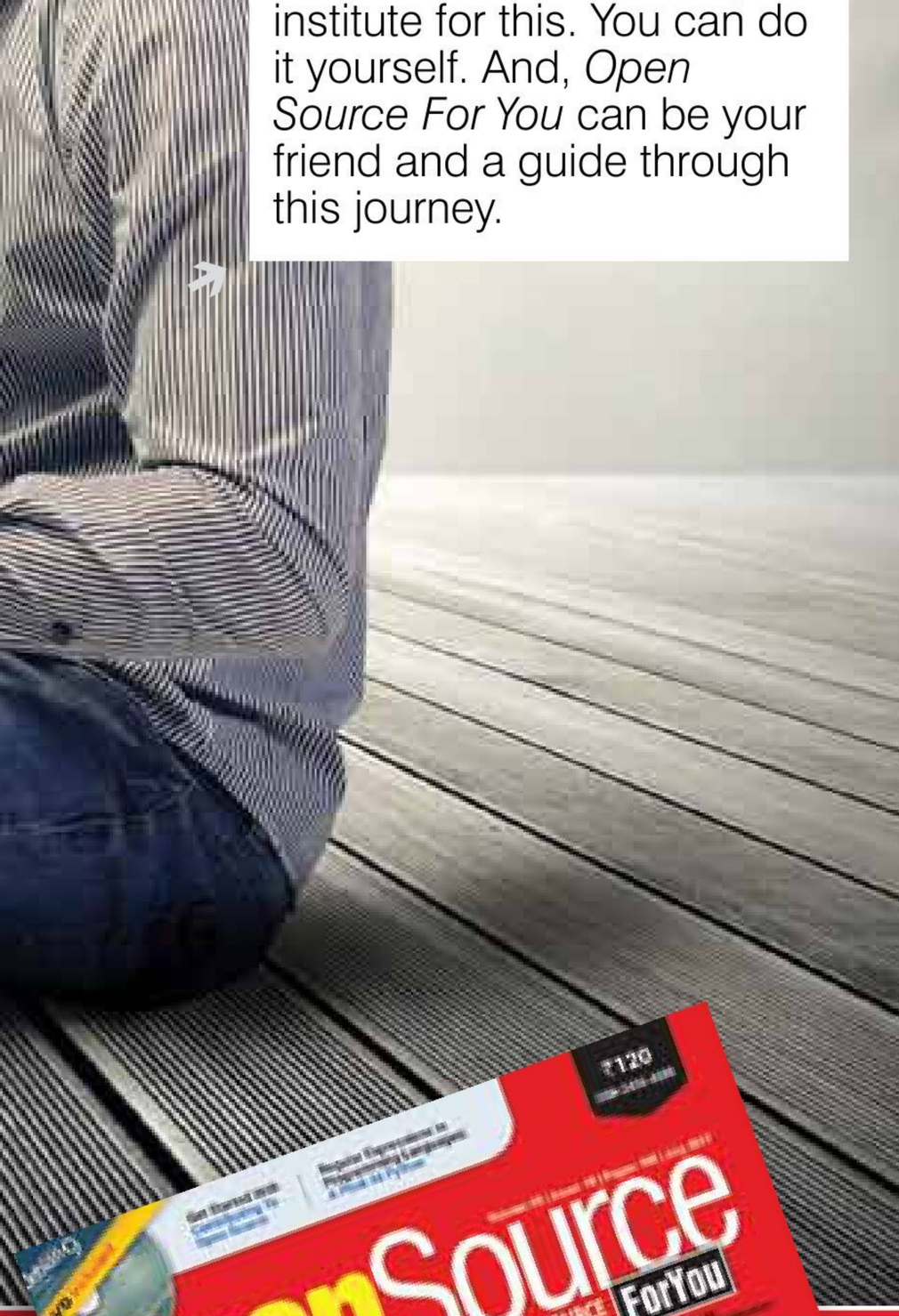
WORK WITH OPEN SOURCE.

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

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

TO READ OUR PRINT EDITION Visit: 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.

EDITOR
RAHUL CHOPRA

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

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

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

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

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

MUMBAI
E-mail: rmowest@efy.in

BENGALURU
E-mail: rmosouth@efy.in

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

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

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

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

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

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

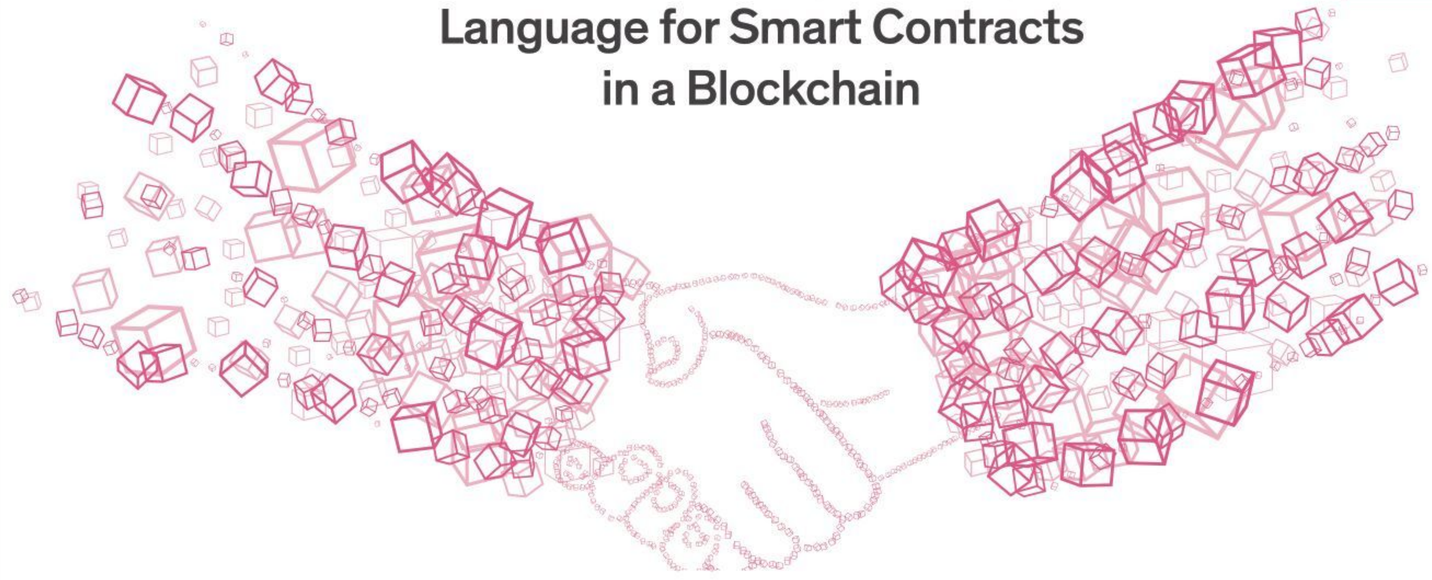
Printed, published and owned by Ramesh Chopra. Printed at Tara Art Printers Pvt Ltd, A-46,47, Sec-5, Noida, on 28th of the previous month, and published from D-87/1, Okhla Industrial Area, Phase I, New Delhi 110020. Copyright © 2021. All articles in this issue, except for interviews, verbatim quotes,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

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

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

CONTENTS

DAML: The Programming Language for Smart Contracts in a Blockchain — 44



The Basic Concepts of Shell Scripting — 48



Machine Learning: Classification Using Python — 51



Using a Machine Learning Model to Make Predictions — 62



Creating a GitHub Action from Scratch — 69



Should You Switch to Serverless Computing? — 86



A Quick Look at Cloud Cost Management Tools — 95



Linux kernel team fixes a major security vulnerability



If you use a Linux distribution on your computer or an Android smartphone, you should apply the latest updates right away since the Linux kernel has been detected and patched with a serious security issue. The vulnerability, known as CVE-2022-0847 and dubbed 'Dirty Pipe', was identified earlier this year by Max Kellerman, a software developer at the Web hosting business IONOS.

Kellerman first became aware of the vulnerability in the Linux kernel since version 5.8 after getting customer complaints about corrupted files, according to a comprehensive blog post. He was able to recognise a pattern, and determine that the cause of the fault was in the Linux kernel itself after an identical problem occurred many times after the first report.

Kellerman notified the Linux kernel team the same day about his discovery, and the team swiftly issued a patch to fix the problem. All impacted Linux versions have received a security update, and Google's Android operating system, which is based on a modified version of the Linux kernel and other open source software, has also been upgraded.

Dirty Pipe can be exploited by an attacker to acquire complete control over impacted computers and cell phones if it is left unpatched on susceptible systems. The attacker would be able to read users' private chats, hack banking apps, and more with this access.

In general, Linux permits each file to have certain permissions for reading, writing, and executing it. An attacker was able to bypass these protection methods due to a flaw in the way memory is maintained for communication between various processes (through so-called pipes).

The Dirty Pipe flaw affects all Linux systems starting with kernel 5.8 and Android smartphones running untrusted apps. According to a recent email from IONOS, while untrusted apps are normally segregated from the operating system as much as possible, the problem might still be duplicated.

Karapace will have Instaclustr support

Database-as-a-service Instaclustr has announced that it will now support the open source Karapace schema registry technology, which was developed by its competitor Aiven. Aiven, situated in Helsinki, also offers database-as-a-service and supports a variety of database and data technologies. It offers managed Apache Kafka as one of its services.

Aiven released the open source Karapace project in 2020, which includes a schema registry for Kafka event streaming data and a REST interface. Instaclustr,

Meta releases open source browser extension for validating code

Code Verify is based on the concept of subresource integrity, in which browsers check the integrity of a fetched file. But it extends the concept to all resources on a website and uses Cloudflare as a trusted third party.

According to Meta, WhatsApp Web users can now utilise the plugin to confirm that the program has not been tampered with or altered. The plugin was created to automatically compare WhatsApp Web code to a cryptographic hash source of truth entrusted to Cloudflare, and to warn the user if any inconsistencies are discovered.

Cloudflare's third-party verification, according to Meta, performs the tests automatically, in real-time, and at scale. Code Verify and the cryptographic hash source of truth on Cloudflare are updated whenever WhatsApp Web is updated.



Code Verify is now available for Chrome and Microsoft Edge, and will shortly be available for Firefox. According to Meta, users may rest assured that their privacy is protected because the extension logs no data or metadata, shares no information with WhatsApp, and has no access to user messages.

"In fact, neither WhatsApp nor Meta will know whether someone has downloaded the Code Verify extension. Additionally, the Code Verify extension never sends messages or chats between WhatsApp users to Cloudflare," Meta said in a post.

ActiveState's Trusted Artifacts ensure greater security

JFrog Artifactory users can now populate their repositories with ActiveState's securely built open source artifacts. Developers get trustworthy, open source packages without being forced to recreate them from scratch, and can now easily manage them with their other artifacts, packages, and software components hosted in JFrog Artifactory. With this additional layer of protection, enterprises can continue to shift security left without disrupting the way developers work.

Open source dependencies provide no guarantees around the security and integrity of the built open source artifacts they provide. Building your own open source artifacts from source code requires significant time and effort but delivers no differentiating benefits to the enterprise.

ActiveState®

Loreli Cadapan, vice president, product management, ActiveState, said: "Open source organisations are making great strides to improve the security of their public repositories, but the reality is that they are still the Wild West where anything goes. Our recent Supply Chain Security Survey results indicate that a worryingly high proportion of organisations continue to implicitly trust these open source repositories. Starting with our secure build service for JFrog Artifactory users, ActiveState is looking to help enterprises overcome these limitations in order to improve the security and integrity of their software development processes."

ActiveState Trusted Artifacts is now generally available.



based in California, said that it is now supporting the Karapace technology by contributing code to the open source project. As part of its own managed Kafka offering, Instaclustr also wants to provide commercial support for Karapace.

On the Aiven platform, Karapace has an increasing number of users, including online travel agency Priceline. Anan Garg, director data engineer at Priceline, said that both Aiven and Instaclustr will benefit from Karapace's backing.

The Karapace project has been designed to be interoperable with the Confluent Schema Registry, according to Aiven. Karapace is now extending that compatibility with integrated support for Google's Protobuf, or protocol buffer, communication format, thanks to Instaclustr's aid.

As part of its managed Kafka service, Instaclustr has been supporting Confluent's Schema Registry and REST Proxy. However, according to Paul Aubrey, vice president of product management at Instaclustr, the company recognised features that customers desired but were not available under an open source licence.

"By now collaborating with Aiven to add support for the Protobuf message format, we have been able to cement the Karapace project as a viable and fully open source alternative," Aubrey stated.

Ingram Micro sets up open source Centre of Excellence (CoE)

Ingram Micro has established an open source Centre of Excellence (CoE) in Asia Pacific to assist partners in capitalising on new enterprise-grade solutions and services. The resource aims to provide the channel with the "knowledge, resources, and applied technology" it needs to better "market, sell, and support" open source offerings throughout the region.



"The move to remote working forced many organisations to accelerate their digital operations plans and innovation cycles to meet customer demands at the speed of need," said Sabine Howest, senior vice president of Digital Operations and Global Partner Engagement at Ingram Micro. "The open source community is making rapid advancements in setting technology trends and addressing problems that are too massive for one organisation to solve. Open source brings upon endless possibilities through collaboration and transparency while maintaining flexibility and accessibility to innovations."

According to Howest, its partners can use the CoE to better identify open source opportunities, particularly for infrastructure modernisation, application development, and digital transformation.

SVA Software partners with Kubermatic

In collaboration with SVA System Vertrieb Alexander GmbH, German software company Kubermatic has partnered with SVA Software, Inc., to bring its Kubernetes management portfolio to the North American market.

The existing relationship between Kubermatic and SVA GmbH is now expanded to enable Kubermatic to enter the US market, and to expand the SVA software portfolio into the Kubernetes and cloud infrastructure management market.



"We are excited to partner with Kubermatic as this is a logical extension of the existing partnership between our parent company and Kubermatic," said Anne-Kathrin Böckle, COO of SVA Software, Inc. "We believe that Kubermatic is a perfect fit for our North American portfolio, expanding our data centre competencies beyond onsite infrastructures in order to support our customers who request flexible and scalable cloud-ready-Kubernetes solutions."

The addition of Kubernetes management/automation solutions to the SVA portfolio consists primarily of two open source products. The Kubermatic Kubernetes Platform (KKP) is a complete solution to address the operational challenges of managing Kubernetes at scale while enabling DevOps teams with a self-service developer and operations portal. Kubermatic KubeOne is a self-service Kubernetes cluster management tool that automates the entire life cycle of single Kubernetes clusters on any chosen cloud, on-premises or in an edge environment.

FTX and Alkemi to support Centre's Verite

Centre – the consortium founded by Circle Internet Financial, LLC and Coinbase to serve as a network governing and standard setting body for blockchain based financial services – has announced a new set of partners for its recently launched set of decentralised identity protocols, Verite. The new partners, crypto exchange FTX and decentralised liquidity network Alkemi Network, will collaborate with Verite to promote broader adoption of crypto payments, decentralised finance, and access to the wider ecosystem.



FTX and Alkemi are joining a collection of partners that have commitments to collaborate on shared decentralised identity standards going forward. These initial Verite launch partners include Algorand, Block, Coinbase, Circle, Compound Labs, ConsenSys, Espresso Systems, Hedera Hashgraph, Ledger, MetaMask Institutional, Phantom Technologies, Solana Foundation, Spruce, and Stellar Development Foundation.

"Today, there is no industrywide agreement over how products and services might interoperate in key crypto finance use cases. Verite is designed to create that consensus," said David Puth, CEO of Centre. "We are pleased that our partners – some of the most influential institutions in the space – share our conviction that Verite's identity standards will create a new level of clarity, privacy, and convenience to everyone transacting in the crypto economy."

Verite is open source and blockchain-agnostic, designed to encourage developers and partners to integrate the protocols into existing systems without introducing single-vendor or anti-competitive dependencies.

Socket Mobile's CaptureSDK now supports Flutter

Socket Mobile, Inc., a provider of data capture and delivery solutions designed to enhance workplace productivity, has announced that its CaptureSDK now includes support for Flutter.

Flutter is an open source framework created by Google for building natively compiled, multi-platform applications from a single codebase. It is engineered to deploy from mobile, desktop, Web, and embedded devices and has quickly gained popularity since its launch. According to the latest figures, Flutter is used by over two million developers, with 500,000 using it monthly. Forty-two per cent of software developers have used Flutter, and it has quickly become the popular cross-platform mobile framework amongst developers, according to a 2021 developer survey.

Socket Mobile continues to expand its toolbox and meet ever-changing software development needs. CaptureSDK now includes Flutter, in addition to native iOS, Android, Windows, C# (Xamarin), React Native, Java and JavaScript, which enables its app partners to choose the development environment that best suits their needs.

The CaptureSDK for Flutter is a single-codebase SDK, meaning that developers do not need to maintain multiple dependencies and complicated logic to target iOS or Android. Developers can even create and apply their iOS and Android app keys in one place. Developers that depend on Socket's CaptureSDK can now utilise Flutter as a powerful tool to develop their application for multi-platform use.

Cloudera and StreamNative develop an open source connector

Cloudera and StreamNative have made an open source connector that integrates Apache NiFi and Apache Pulsar. The connector will be included with the Cloudera platform. NiFi and Pulsar can be combined to create a cloud-native streaming data platform capable of ingesting, transforming, and analysing massive amounts of data.



Cloudera's team includes some of the original Apache NiFi developers, whereas StreamNative was founded by the original Apache Pulsar creators. NiFi was created from the NSA's 'Niagara Files' technology and made available to the Apache Software Foundation via the NSA Technology Transfer Program. NiFi is a flow-based programming visual tool that can be used to create data flows that move data from one technological platform to another, such as databases, cloud storage, and messaging systems.

NiFi provides data provenance and traceability at the event level. The NiFi platform comes with over 100 pre-built processors. Apache Pulsar is a cloud-native, distributed messaging and streaming platform that was developed at Yahoo! and is now an Apache Software Foundation top-level project. To provide long-term stream storage, Pulsar employs a replicated distributed ledger.

Red Hat Survey: Businesses are more likely to opt for open source vendors

According to Red Hat's recent State of Enterprise Open Source Survey, 82 per cent of decision makers are more likely to choose an open source vendor.

The top reasons for this are that they are experienced with open source methods and help maintain healthy communities (both indicated by 49 per cent of respondents), can influence feature development (48 per cent), and are more likely to be effective in the face of technical obstacles (46 per cent).

During the pandemic, open source development has got a boost, and 95 per cent of the more than 1,200 IT executives polled said it's important to their enterprise infrastructure strategy and is steadily replacing proprietary software as a source of innovation.

Enterprise Open Source (EOS) now represents 29 per cent of all workloads. In two years, this figure is expected to climb to 34 per cent, while proprietary software will decline from 45 per cent to 37 per cent.

According to the survey, 77 per cent of respondents now have a more favourable opinion of enterprise open source than they did a year ago. Furthermore, 55 per cent of respondents cite the ability to use well-tested code for in-house applications as the primary reason for choosing open source.

Google's Jigsaw unit releases Harassment Manager for journalists

Google's Jigsaw unit is providing the source code for Harassment Manager, an open source anti-harassment tool. The application, which is aimed at journalists and other prominent figures, uses Jigsaw's Perspective API to allow users to sort through potentially harmful remarks on social media networks like Twitter. In June this year, it will be released as source code for developers to build on, followed by a functional application for Thomson Reuters Foundation journalists.

Harassment Manager can use Twitter's API to combine moderating features such as concealing tweet replies and muting or blocking users with a bulk filtering and reporting mechanism. Threats, insults, and profanity are among the elements used by Perspective to assess the "toxicity" of messages. It organises messages into queues on a dashboard, allowing users to respond to them in bulk rather than individually using Twitter's usual moderating tools. They can opt to blur the text of the messages so they don't have to read each one individually, and they can search for keywords in addition to using the automatically generated queues.



On International Women's Day, Jigsaw released Harassment Manager, emphasising that the tool is especially relevant to female journalists who endure gender based abuse, "journalists and activists with large Twitter presences," as well as non-profits like the International Women's Media Foundation and the Committee To Protect Journalists. "Our hope is that this technology provides a resource for people who are facing harassment online, especially female journalists, activists, politicians and other public figures, who deal with disproportionately high toxicity online," the Jigsaw team posted.

Harassment Manager, unlike AI-powered moderation on platforms like Twitter and Instagram, is not a platform-side moderation feature. It is a sorting tool for dealing with the sometimes overwhelming volume of social media comments, and could be useful even for individuals who are not journalists.

DENT 2.0 NOS available for download

The DENT project is a Linux Foundation-hosted open source network operating system that uses the Linux kernel, Switchdev, and other Linux based projects. DENT 2.0 is now available for download from the project's website.

The 'Beeblebrox' edition includes critical capabilities used by distributed organisations in retail and remote locations, as well as a secure and scalable Linux based network operating system (NOS) for disaggregated switches that may be deployed at the edge. DENT provides a smaller, lighter NOS for use at an enterprise network's small, remote edges.

To accommodate a larger community of enterprise customers, DENT 2.0 offers secure scalability with Internet Protocol version 6 (IPv6) and network address translation (NAT). Power over Ethernet (PoE) control is also included, allowing for remote switching, monitoring, and shutdown. Connectivity of IoT, POS, and other devices is extremely advantageous to retail stores, which are DENT's early adopters. DENT 2.0 also has traffic regulation, which helps to prevent circumstances that cause the CPU to overheat.

Kerala's CM releases FOSS based operating system suite

The improved KITE GNU/Linux operating system (OS) suite – the FOSS based (free and open source software) OS created by the Kerala Infrastructure and Technology for Education (KITE) – was recently released by the state's chief minister Pinarayi Vijayan for deployment in new laptops in schools.

This operating system package can be used as a comprehensive computing platform by students and teachers in schools, on personal computers, at government offices and DTP centres, and by software developers.



The new package includes all of the Ubuntu OS's most recent updates. It also includes a large number of FOSS based apps not found in the Ubuntu 20.04 repository. The majority of the OS suite's apps have been updated to their most recent versions and tailored to the school curriculum, particularly for the higher secondary and vocational higher secondary portions.

The new OS suite includes a wide selection of Malayalam Unicode fonts, DTP graphics picture editing tools, sound recording-video editing 3D animation packages, IDEs for programming, database servers, desktop versions of mobile apps, and more. The OS also includes internationally acclaimed FOSS programmes such as GeoGebra and GCompris, which aid in the learning of various disciplines through the use of technology.

Microsoft launches open source forensics solution for Mikrotik routers

Microsoft's Defender for Internet of Things Research Team and Microsoft's Threat Intelligence Centre have collaborated to create an open source forensics solution for the Mikrotik routers that have been compromised to function as proxy servers for the Trickbot malware gang.

The researchers discovered that attackers with several means to access Mikrotik routers were able to reroute traffic to and from the Trickbot command and control servers by using instructions particular to the Latvian vendor's RouterOS operating system (preceded by the "/" character). Administrators can get device version identifiers and link them to Common Vulnerabilities and Exposures (CVEs) indexes using Microsoft's routeros-scanner tool.

cPanel to provide full support for Ubuntu LTS 20.04

cPanel LLC, a WebPros portfolio company, will provide full support for Ubuntu LTS 20.04 with cPanel and WHM version 102.

In the winter of 2020, CentOS 8 had announced that it would be reaching its end-of-life in December 2021. This news was somewhat unexpected in the open source and Web hosting worlds. In response, cPanel has been working towards providing solutions that will benefit its partners and customers, and expand options.

Aila Power, VP of product development at cPanel, said: "Ubuntu is a solid Linux distribution with a great track record of security and innovation. I'm really excited that cPanel and WHM are now available on this distro that is the first choice for many when they build a new application. Our collaboration with Ubuntu will give cPanel and WHM users more open source Linux distribution architecture options within their current infrastructure while providing a more robust business suite of solutions to help customers scale and expand their business hosting operations."

"Ubuntu has long been the platform of choice for large scale compute estates and Canonical is delighted to partner with cPanel to mitigate the impact on long term stability and predictability brought about by the changes to CentOS," shared Mark Lewis, VP of application services, at Canonical | Ubuntu.

PyGears, the open source chip design language based on Python released

An open source chip design language based on Python and optimised for machine learning techniques has been released by a consortium of researchers in the United States.

PyGears is a new hardware description language developed by a group at the University of California (UCLA) in the United States. It is based on the concept of reusable components and high-level Python constructs in agile chip design.

The UCLA Samueli School of Engineering claims that by employing it in the VLSI Signal Processing course, it has begun to develop a community that observes hardware through software paradigms, with the goal of expediting hardware design and verification by using more agile and composable modules.



"Chip design is getting more complex, with increasing costs and longer development cycles," stated Dr Dejan Markovic, electrical and computer engineering professor at UCLA. "PyGears addresses these challenges by adopting a modular build of hardware out of reusable components, which are built by a community of like-minded contributors. This works very well for small and large engineering teams alike. Hardware design and verification is based on the Python environment, reducing the barriers between software and hardware domains. The vision is to enable software people to code hardware."



Trickbot, which is based on the previous Dyre malware, is one of the world's most hazardous botnets and is frequently used to transmit ransomware payloads. Microsoft security professionals worked with law enforcement agencies throughout the world to shut down Trickbot's infrastructure, resulting in the arrest and extradition of some of the malware's creators to the United States to face charges.

Firefly introduces ValidIaC

Cloud asset management solution Firefly has announced the release of a new open source solution called ValidIaC, which enables DevOps and cloud teams to validate, secure, and map Infrastructure-as-Code (IaC) with an all-in-one browser based tool.

Inspired by Komodor's open source project ValidKube, which combines the best open source tools to help ensure Kubernetes YAML best practices, Firefly has built a sister project for the purpose of Infrastructure-as-Code.

ValidIaC combines several open source projects and tools to allow developers to check their Terraform HCL code without changing context and from within the same workflow.



"Infrastructure-as-Code is changing the way modern organizations are managing their infrastructure. With ValidIaC, developers can move faster but still have confidence when it comes to security, compliance, cost, and reliability," said, Ido Neeman, CEO at Firefly.

ValidIaC is an open source tool, so community contributors can add more tools or capabilities that will help them and the community. Firefly announced that it'll soon introduce support to more Infrastructure-as-Code tools, such as Pulumi, Cloudformation, etc.

Zenoss discontinues Zenoss Community Edition

Zenoss Inc., has announced that it is sunsetting Zenoss Community Edition (previously called Zenoss Core), a free, on-prem monitoring tool the company made available for over 15 years. Since the launch of Zenoss Cloud in 2018, which saw a 202 per cent increase in annual recurring revenue over the past two years, the focus has gradually transitioned from crowdsourced development of extensions for Zenoss Community Edition to community development of cloud tools.

Zenoss Community Edition version 1.0 was released in November 2006. Since then, the product has been downloaded millions of times and became the *de facto* open source monitoring tool for companies of all sizes across all industries.

Open source trends for 2022 in Southeast Asia

Ashnik, an open source solutions provider in Southeast Asia and India, recently released its survey with participation of over 160 cross-industry organisations from the region. The survey captures the growing role of open source, as well as the key digital transformation initiatives and technology trends of 2022.

"Open source has been a big enabler in the areas of infrastructure automation, cloud, real-time analytics, security and more, making its adoption a key focus for top global organizations. The survey is the voice of several enterprises reflecting the growth of open source, top initiatives driven by it, the rising market share for its services and support, and more – giving a multidimensional understanding of organizational goals," said Deepti Dilip, director-marketing and strategy at Ashnik.



The survey indicated that about 82 per cent of organisations today rely on open source to 'drive innovation' within them; 100 per cent voted 'ease of adoption' as the best feature of open source. Over 95 per cent of respondents cited that open source offers 'higher security' over proprietary software, 71 per cent preferred to consume open source technologies in the cloud through a SaaS model, while 53 per cent picked a hybrid cloud solution (public cloud + data centre) as their preferred deployment platform. Seventy-four per cent participants indicated the adoption of open source databases is set to increase in 2022, with over 59 per cent indicating an increased usage of community open source databases specifically.

Fifty-seven per cent of respondents said that the lack of internal open source skill-sets and manpower was a key challenge in their organisations, while 97 per cent desired to have access to service providers and consultants for the success of their projects using open source.

The top open source database technologies to see an increased adoption were PostgreSQL and MongoDB, followed by Redis and Elasticsearch; and the top Kubernetes platforms of 2022 were AWS EKS and AKS.

Wipro joins the governing board of OpenSSF

Wipro Limited has joined the governing board of the Open Source Security Foundation (OpenSSF) to help combat the growing danger to the software supply chain.

The OpenSSF is a cross-industry organisation based at the Linux Foundation that brings together the world's most important open source security initiatives to help identify and fix security vulnerabilities in open source software, as well as develop better tooling, training, research, best practices, and vulnerability disclosure practices.



Wipro's open source expertise will join other members in determining direction through the foundation's governance and working committees, in addition to developing and sharing best practices for safe coding and software components for the projects under the OpenSSF umbrella.

"We are thrilled to now count Wipro as a key strategic partner in the OpenSSF community," said Brian Behlendorf, general manager, OpenSSF.

The Open Group India Awards 2022 to be held in Washington

The Open Group India Awards 2022 will be hosted virtually on July 27, 2022, in Washington D.C., in conjunction with The Open Group International Event. Since its inception in 2017, The Open Group India Awards event has honoured

Strapi introduces TypeScript support

Strapi, an open source headless content management system, has launched the beta version of Strapi 4.2 with TypeScript support, Dark Mode, and In-app marketplace at the second annual StrapiConf, the free global user conference for the Strapi community.

TypeScript, one of the most requested features on the Strapi roadmap, now makes the Strapi codebase more reliable, accessible, and easy to refactor for developers.

"When we developed Strapi v4, we anticipated TypeScript support by introducing an internal typing system," said Alexandre Bodin, CTO, Strapi. "The new TypeScript support relies on that, but it will also be very useful for schema and data migration. This brand new version also introduces a dark mode option for those who prefer dark interfaces. This new functionality also sets the foundations for more advanced and custom themes in the future."

StrapiConf brings together more than 4,000 attendees and 20 sponsors from the Strapi community to learn, contribute, and collaborate on the future of Strapi.

At StrapiConf, the company also shared details about a new product called Strapi Cloud, which is based on a platform-as-a-service approach to enable a seamless deployment process without having to compromise on Strapi's unique customisation capabilities.

Google to contribute US$ 3 million for Data Transfer Project over the next five years

Google has committed US$ 3 million over the next five years to the Data Transfer Project, as well as hundreds of hours of its engineers' labour. In 2018, Google, Facebook (now Meta), Microsoft, and Twitter announced the Data Transfer Project, which focused on developing tools to assist consumers "transfer data directly from one service to another, without needing to download and re-upload it."

According to Google, the fresh financing will be used to create open source frameworks that enable more forms of data transfer and allow more firms and organisations to join the project. People who don't have high-speed Internet, unlimited mobile data plans, or a personal device with extra storage may find data transfer difficult.

"In 2018, we launched the Data Transfer Project (DTP), an open source collaboration with Google, Apple, Meta, Microsoft, Twitter and SmugMug to simplify data portability for people around the world. On average, we see 8.2 million exports per month with Google Takeout and in 2021, more than 400 billion files were exported, which has doubled since 2019," the company has stated.

Unlike typical ways of moving files from one provider to another, which necessitate good broadband or the use of mobile data plans, the project allows users to simply authorise a copy of their data to be moved to a new service without having to first download it to a personal device. This allows customers to try new services without having to worry about running out of storage space.

"We will also continue to improve our own tools like Google Takeout, including adding new ways to move your files to different services with DTP technology," said Google.



organisations and teams across Southeast Asia that have excelled in using open source software, standards, and best practices in enterprise architecture, IT management, cyber security, and digital transformation.

Nominations are accepted from both government and private sector organisations. Winners and other selected submissions may be invited to speak at the awards ceremony in order to promote their company as a pioneer and leader in their field to a larger audience.

Kava adds EVM support with alpha launch of Ethereum Co-Chain

Kava, an open source Layer-1 blockchain, has added EVM smart contract support with the alpha launch of its Ethereum Co-Chain. The launch adds EVM developer support to the network, enabling developers and dApps from the Ethereum ecosystem to build and deploy on Kava.

Over 15 protocols will be deploying to the closed test net of the Ethereum Co-Chain as part of the Kava Pioneer Program, including Beefy Finance, AutoFarm, and RenVM. These projects will test the interoperability between Kava's Ethereum and Cosmos Co-Chains prior to their mainnet launch.



Through its co-chain architecture, the Kava Network merges the developer power of Ethereum with the speed and interoperability of Cosmos in a single, scalable network. This allows Cosmos and Ethereum developers to build on one chain to access the users and assets of both ecosystems.

"Ethereum is still where the vast majority of developers and protocols are, but Cosmos is growing fast and it offers so much more in terms of scalability and interoperability. Bringing the best of both ecosystems together on Kava just makes sense for our goal to add 100 protocols this year," said Scott Stuart, Kava Labs CEO.

Google Java App Engine standard now available

The Java source code for the Google App Engine Standard environment, the production runtime, App Engine APIs, and the local SDK have been made public by Google.

Google App Engine was first published in 2018 with the goal of making it simple for developers to install and scale their Web applications. Many languages are presently supported by App Engine, including Java, PHP, Python, Node.js, Go, and Ruby. Java developers may use Java 8, Java 11, and Java 17, as well as other JVM languages like Groovy and Kotlin, to deploy servlet-based Web applications. Spring Boot, Quarkus, Vert.x, and Micronaut are just a few of the frameworks that can be used.

Developers may now operate the entire App Engine environment wherever they choose, including on-premise in a data centre, on alternative computing platforms, or even Google Cloud like Cloud Run, thanks to the open source of the App Engine Standard Java runtime.

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

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)

"The future is SaaS plus open source"

Jina AI provides cloud-native neural search solutions powered by state-of-the-art AI technology. The company says its neural search framework called Jina empowers anyone to build SOTA and scalable neural search applications in minutes. In a conversation with ***Abbinaya Kuzhanthaivel, Pradeep Sharma, senior developer relations expert, Jina AI,*** touches upon his open source journey, the range of skills needed in this industry, the challenges in the open source world, how open source helps to speed up project development, and more.



Pradeep Sharma,
senior developer relations expert, Jina AI

Q. Can you tell us a bit about yourself and why you chose to work on open source projects?

A. I am working with Jina AI, the cloud native neural search solutions provider, as a developer relations expert for building relationships with the developers of our open source projects.

I graduated from IIT Roorkee in 2012 as an electronics and communications engineer, and always had a keen interest in programming. I joined a FinTech startup as a software engineer and built a lot of projects there. When one of my projects worked really well, I built my own startup and worked in it for about two years. But then I decided to shut it down and work for developers, especially those from remote rural areas. I could see that people from Tier II and Tier III cities do not get the kind of opportunities people in metro cities get. I started building a community of such people in 2016, which now numbers 15,000. While programming was my prime skill, I had to develop many more skills like communicating with the target audience as well as some skills in sales that could help me run an organisation.

Q. Is Jina AI your first venture into open source?

A. No. I built my own open source project back in 2016-17. I started teaching people about JSON Web Token (JWT) and what is JSON Web authentication. I did this through interactive demos. I started this developer community, and then we needed a package for encrypting some of our data. So I created an NPM package and published it on GitHub.

I had organised an online conference called the Git Commit Show in the pre-Covid days. I open sourced all the tools we had used to organise that online conference, and got a huge response from the audience. That built my confidence and indicated that open source is a great way to engage and collaborate with developers.

Q. What is meant by neural search?

A. When you search for something on Google with a keyword, it will probably give you results based on that or something similar to it. But searching by keywords is not the only way.

We use neural networks to do a semantic search, which makes your experience so much better. It doesn't matter what you provide – keywords, images or videos — it will give all the relevant results. This search is based on the examples you provide, and it learns from that. So the next time you search for something, it will just look into the repository that you have built to give you the results.

Q. What are the challenges you have faced in your open source journey?

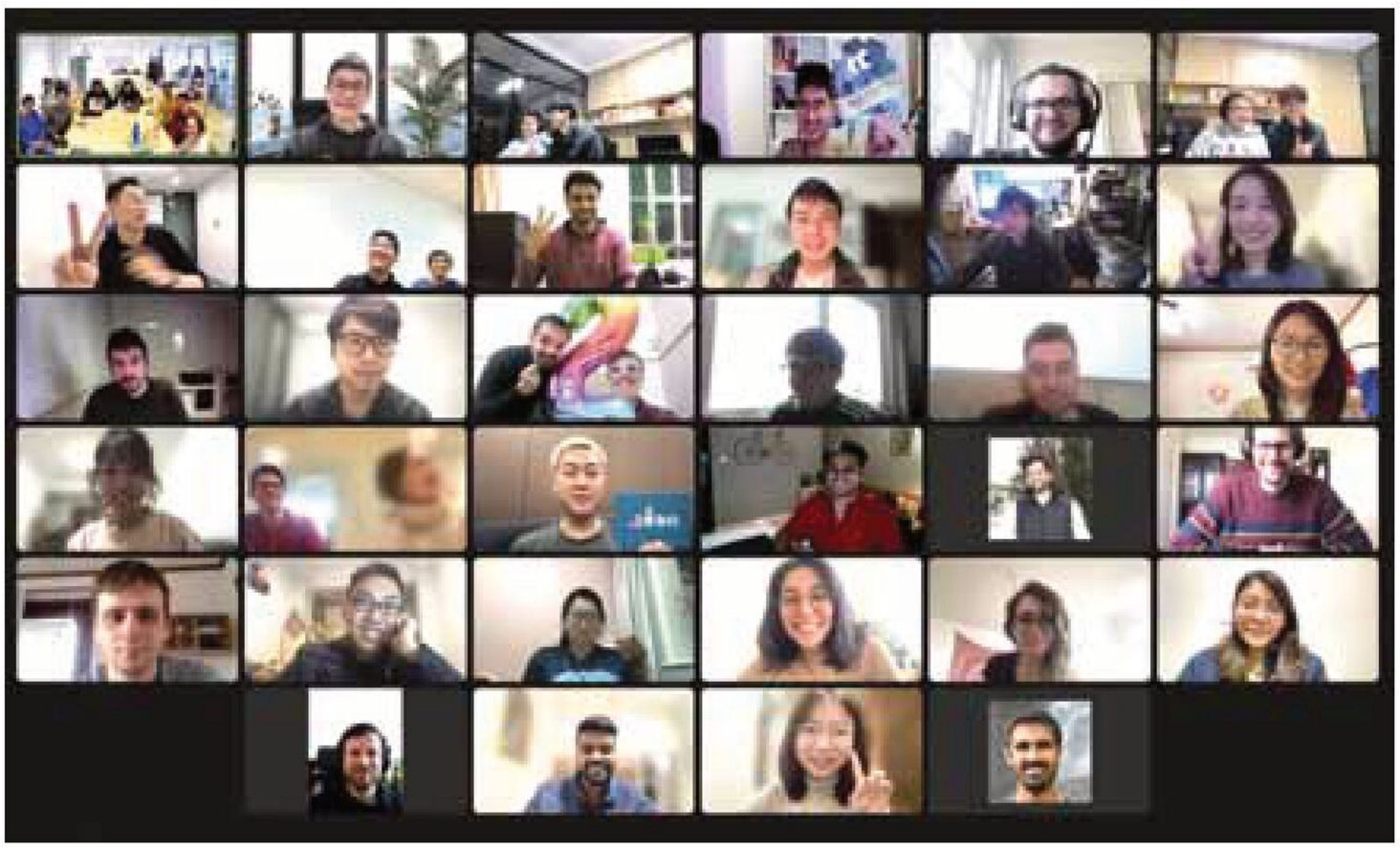
A. Courage is needed to be more open and transparent, and to share your code with people. One also needs a change in mindset and loads of confidence. I often felt that my code may not be perfect, and was not sure if I should share it with people. There is also a fear about how secure the code is.

However, when I had organised an open source event, people like Daniel Stenberg and Sir Tim Berners-Lee encouraged me to just go ahead and open source the code, since people would support the project and also point out vulnerabilities and help to fix

them. The project would become more secure that way.

I remember a recent incident where community members and developers teamed up for an online meeting to publicly discuss queries and concerns on Jina AI. We started this call and someone Zoom-bombed us by playing very loud music along with verbal racist abuse. This was a pretty big challenge. Thankfully, our community supported rather than blamed the organisers.

Ensuring the safety of my community members is a mindful task, but the good generally outweighs the bad.



Q. How can open sourcing a project speed up its development?

A. Open sourcing does help to speed up the project. First, once you find members, you get the motivation to take the project forward faster, and that's what happened with me at Jina. There may be people who challenge you every day, and tell you that they have found some problem with the project and it's not working. So you need to build the motivation, internally and externally. But then there are those who ship their code quickly, helping the project move forward.

More importantly, collaboration works magic mysteriously. When you share the project with the world, people start using it, and then they come up with their own ideas and notions as to how it can be used and build something further out of it. They may build something that is a little different from what you built, and someone else may even make it better. And this eventually grows your project at a very exponential scale. Such growth cannot happen if you are working alone, limiting that code to a certain number of people in your team.

The best example I can share is from Jina itself. The first version took us about one year to release and the next version was released in three months. Our contributors have grown to maybe around 200 or more.

> "If you are a developer, this is the time to think about open source and the ways in which you can contribute to it and be more involved in that space. Just keep learning and build projects."

Q. Do you mean to say that the number of people working on it can also impact a project's speed?

A. It is about the number of people who are engaging with the project and not necessarily only the ones contributing code to it. At the time of the Jina 1.0 release, we had 2000 stars on GitHub. This figure has reached 13,500 now. So these many people are sharing their ideas on Slack, reporting issues or helping out with other stuff that they find new.

For instance, the Jina 1.0 version puts structure to the data, creates a pipeline to operate on this data, and then eventually makes it available to the audience for search operations. The version looked very simple initially, but we figured it was complex. So we decided to break it down into three separate projects and saw an increase in its usage. The growth of Jina has been exponential in 2021.

Q. How does one sustain open source projects? Any tips?

A. As a project lead, always remember you have to lead the project. Do not expect people to come and start contributing on the first day. You have to put in the effort as a leader and have to move the project forward. There will be contributions to the code or in fixing vulnerabilities, but there is no one who will give so much support that the maintainer or lead does not need to think about the project.

You can think of how to make people come in and contribute to your open source project. A couple of things have worked well for me. When you are passionate about your project and people recognise that, the chances that they will get inspired and start supporting you are very high.

The other thing is to keep your project simple. If people find your project really complex to understand, they are doubtful about whether they can contribute to the code. So make your project easy to contribute to, and cut down the clutter in terms of code and ideas.

Q. Who do you look up to as inspiration?

A. Tim Berners-Lee, the creator of the Web, is my first inspiration. I keep communicating with him and sharing emails, and I do get a response once in a while. The other person is Daniel Steinberg, the author of cURL. He built that project, which is installed on every computer in the world. So that is something that is very inspiring. He is very transparent and shares his experiences on his blog. I seek advice from him, and I think he is among the best persons to follow in the open source world.

I would encourage everyone to attend events, as that is where we can come across the achievers in the open source industry. That really helps to build your network and push projects forward.

Q. Any nervousness during your open source journey, and how did you overcome it?

A. Oh, yes! That happens almost every once or twice a year and it's absolutely fine to accept those situations. I had to shut down my startup when I realised it was not working out. That was really difficult, but it kept me pushing to think about what exactly I wanted to do.

I moved on to working with Jina. I now understand that one need not go by the trends out there, but rather be more focused on what you really want to do for the long term. We have a huge community of 15K ML engineers at Jina now.

Q. What's your message to all the developers out there?

A. I think people should be more aware of open source as I can see a rapid transformation in the industry. I strongly believe that any software that is proprietary right now will soon have an open source alternative — if not this year, then maybe in just a couple of years. I foresee a future built on open source.

If you are a developer, this is the time to think about open source and the ways in which you can contribute to it and be more involved in that space. Just keep learning and build projects. And the bigger question to address now is building sustainable open source projects. There is a sweet spot for developers to build open source software in a transparent manner and make it sustainable by producing a software as a service (SaaS) version of it. For those who do not want to go through the struggle of maintaining software, SaaS could be the best solution. This may be a bit early to state, but I am pretty confident that most of the projects we see in open source will really move to this kind of model to make it sustainable. And that's the future — SaaS plus open source.

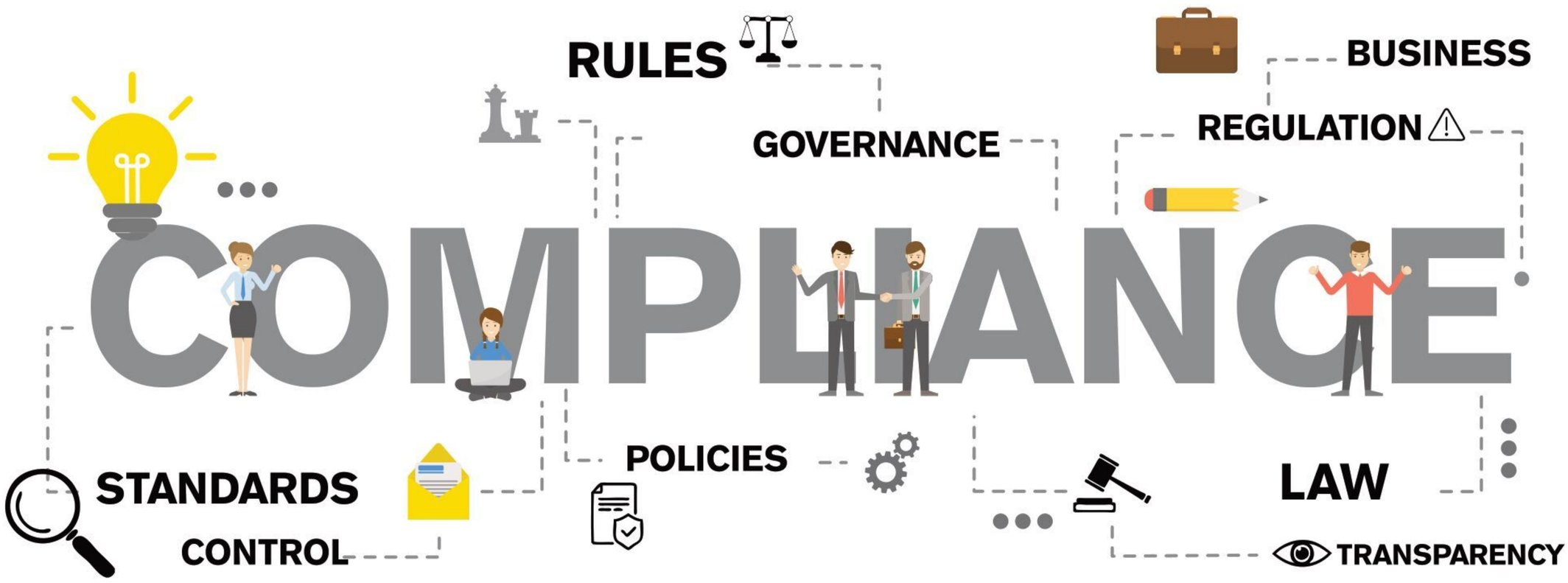
Q. Are you planning any expansion and what are the skills you are looking for?

A. At Jina AI, we are currently hiring for engineering roles. For my team, I am looking for people with technical writing, community development and event management skills END

OSFY Magazine Attractions During 2022-23	
Month	**Theme**
March 2022	Security, Network Management and Monitoring
April 2022	Open Source Programming (Languages and tools)
May 2022	DevOps Special
June 2022	AI, Deep learning and Machine Learning
July 2022	Database management and Optimisation
August 2022	Cloud Special: Everything from management to implementation
September 2022	Mobile/Web App Development, Optimisation and Security
October 2022	Blockchain and Open Source
November 2022	Open Source and IoT and Edge
December 2022	Data science, Data security, Data Management,Storage and Backup
January 2023	Docker and containers
February 2023	Open Source on Windows and Best in the world of Open Source (Tools and Services)

Are You Ensuring Open Source Compliance? If Not, It's Time to Do So

Open source codes, libraries and designs are all available for use against a licence. As a developer and a user, you must make sure you are aware of the various compliances required, and also ensure that your software complies with these.



Open source compliance is a process by which open source users, integrators and developers satisfy the licence obligations implicit in their open source software components. A well-designed open source compliance process will help organisations protect their intellectual property.

First step to compliance: Awareness

A lot of companies do not consider themselves as open source companies. And because of that, their business leaders don't believe they need to manage open source. But, in reality, all software developed today consumes open source components, libraries and frameworks. This use of open source carries a few risks, which are then passed on to customers. Indian developers have contributed hugely to a lot of open source projects like Soda and HarmonyOS on a global scale. Open source compliance is today mandatory to ensure that there are no issues in the use of these projects.

Sriharsha Narayanam, lead architect, Huawei Technologies India Pvt Ltd, points out three scenarios where one must consider compliance. The first is the open source project that is developed and hosted, where the highest responsibility and obligations lie with the creator. The second is when developers contribute to third party open source projects, which comes with a lot of shared responsibility. The final scenario is when we use open source projects. "Give enough attention to each of the scenarios — be it with respect to the tooling or the guidelines. The key responsibility is to create awareness among all the members joining open source projects," says Narayanam.

Martin Callinan, director, Source Code Control India Pvt Ltd, a company that provides services to manage licensing, says, "Training programmes should be provided about what needs to be managed and how to manage it. The companies can be helped in making policies. For instance, the organisation may have intellectual property (IP) value in the software,

for which the source code cannot be shared. It is advisable to create a guidance for developers about what to avoid coming into their code that will be shipped to the customers. It must also include services on security vulnerabilities in the components used. Many companies can't actually identify what open source is being used. Such firms need help to refer to software composition analysis tools, which can identify the licences associated with components and vulnerabilities," he says.

Balakrishna Mukundaraj, associate project manager, Bosch Global Software Technologies Pvt Ltd (BGSW), also feels it is important to develop training, policies and other processes that are necessary to achieve open source compliance across the organisation. BGSW is among the first companies to provide compliance audit services to all the different business units of Bosch worldwide. "The consultation services begin during design — companies come to us with queries about licensing at the stage of procuring components. The organisation level training is vital even for any tools that have been procured or developed in-house," he says.

"Education is the core foundation of open source compliance. An open source awareness programme for onboarding graduates will help to figure out the roadmap to be implemented within a company. At Bosch, we took this up seriously and started giving more time towards the training, which has proved very useful," he adds.

Along with training and awareness, a little knowledge about the common misnomers and challenges can help developers improve their understanding of open source compliance. The Linux Foundation offers two courses for this:

- Introduction to Open Source License Compliance Management (LFC193)
- Open Source Licensing Basics for Software Developers (LFC191)

A few myths

The software is free: The most common misunderstanding in open source is with the term 'free'. "It actually means you are free to use it and do whatever you want within your project as long as you meet the obligations of the licence," says Aykut Avci, global license compliance manager, iText Software.

As there are numerous permissive, non-permissive and copyleft licences like Apache License (permissive licence), Massachusetts Institute of Technology License (permissive licence), General Public License or GPL (non-permissive and copyleft licence), Affero GPL (non-permissive and copyleft licence) and more, it's quite difficult for a developer to know the obligations of each one of these. "The open source you use may be free, but there are different business models that can allow a certain part of it or certain functionalities to be open, and the rest is for commercial use. People frequently confuse the two. For instance, iText has a dual licence – if you use the open source copyleft licence, you will need to make the source code of your derivative work available to users, but you can purchase a commercial licence to take away the copyleft obligation," he adds.

Open source software has good quality: We expect an organisation to have a very high quality of open source. But remember, there is good and bad quality open source. The update history and commits of open source projects are a good reflection of the health of an open source component. The community is not going to check that.

Narayanam says communities do have strong pipelines and checks, but they may not work to achieve quality. Hence, always set your standards high for the open source project and the delivery project, with respect to the quality, cyber security and tooling.

Callinan says it is a myth that if it is managed commercial software, it does not have much risk. It is also a myth that open source communities will fix everything. Some projects are managed only by one person. To check the quality of a project, look at the activity around the components, library or the package you want to bring into your software. There are tools that can reveal data about the number of commits causing a hidden risk for companies, such as the recent log4j vulnerability. "Always check how well-supported the product is," he adds.

The OpenChain Project of the Linux Foundation is a great example of how trust can be built into the complete supply chain of software.

Compliance issues are being taken care of: According to Narayanam, when developers are collaborating on an external open source project, the host may not take care of the compliance. So, how does one ensure compliance? It's best to have a common understanding of the copyright statement right from the beginning when you are contributing to or hosting a project. The advice of a legal team is mandatory.

From a quality perspective, give details on what kind of tooling you have used to ensure compliance when the code is checked. Also, place checks and balances on what level of cyber security you want to ensure.

The engineering team, developers, legal team and IP team will all need to be involved in compliance.

Licensing can be ignored: There are many different types of licences and their versions. Tools can be used to identify the components in the code, but unless you code in, they generally don't provide interpretation. As this becomes confusing, people tend to avoid using them.

"We must look at the business risk as opposed to working just with developers. Developers shouldn't have to spend time interpreting licences. Companies must put guidance in place. Also, start working with a company by attending education programmes

to get some understanding across the business about the different types of licences and implications you need to manage, and what tools you need to put in place," Callinan says.

He points to the famous case of Tesla, where the company was forced to share the source code of its navigation system. Investors and companies need to evaluate and find if there are any risks in the code being used.

Licences are not ambiguous: Mukundaraj says it is very difficult to explain licence ambiguity to your development team, whether it is internal or external. It's even more tedious to figure out whether your vendors have also complied with the different licences. Licences get pretty ambiguous within selected categories at times, and you don't know whether you are compliant with them or not.

Certain licences can be used only for educational institutions and not for a commercial organisation. But sometimes developers think it is fine to use them for their proof-of-concept within the company and purchase a commercial licence later.

"You are just not allowed to do that, because the licence clearly says that if you want to use the free version, you either have to be an education society or a non-profit organisation. Otherwise, you cannot use it even for proof-of-concept or proof-of-experiment. Stay in touch with your lawyers, so that you don't make your own choice of licences,'' he says.

"Always include security in your checklist as resolving any breaches will cost you more," Callinan warns.

Also look for the people who are enforcing open source licences. Traditionally, this would be the community's free software foundation or a freedom law centre. But now it turns out that competitors, individuals, developers and corporates are enforcing these. For example, in the case of CoKinetic Systems Corporation versus Panasonic, the former claimed damages for a hundred billion US dollars and Panasonic was forced to comply.

"If we find text in a company's code without a commercial licence, they may not be sharing their source code in obligation to some GPL licence. I would look at that as a risky investment. OpenChain can be used to check for mergers and acquisitions. Benchmark a target company, audit its maturity with respect to managing issues, and then you can decide if it is a risk investment or not," says Callinan.

Check out these tools and tips

There are great projects like SPDX, which is an ISO 5962 standard that can be used by companies. Open source tools and projects should be used for compliance and detection of libraries, snippets, licensing and obligations.

Say, for example, if you look at codes on GitHub, the licence information is never consistent. Some people use copyright statements, Apache licence, and so on. This makes it hard to manage all those different licences coming into your code. In such cases, it's best to look for an SPDX software package or Data Exchange kind of projects. Callinan says, "The idea of SPDX is to drive consistency of documentation of the licences posted, both in human and computer readable formats. So when you consume software, you can easily identify what licences are in the code. And when you ship your software, you can easily communicate about those licences, which makes it easier to manage the software."

Narayanam suggests FOSSology be used to understand licensing copyright compatibility. He says, "This binary scan tool will help to find out all the binaries and the corresponding source code that can be risky. And with respect to security concerns, fuzzing angle, peach fuzzer or a scan burp suit can help. FindBugs and Sonarqube can support quality. So, if some project is hosted on GitHub, we can incorporate a lot of tools if it is completely open source."

Commercial open source software can be used outside the scope of the open source licence. Avci says there are cases where a developer uses one version of a software in a project that comes with an Apache licence. When this software is upgraded to a new version, the developer assumes the licence has been complied with already. But this may not be the case. The compliance check is ongoing. "An open source project like iText has had three major releases in the past and the latest has a new AGPL licence model. A user compliant with the earlier versions just cannot overlook the new licence," says Avci.

Since it is not easy for small organisations and enterprises to consult lawyers every time, Avci recommends a few simple checks to avoid risking your business. In large enterprises, open source software is often implemented with the help of external software providers. These enterprises may not be aware of what is in the software they receive from a third party software vendor. So it is a good practice to check the software bill of materials (SBOM) as it will tell you if the software given to you is compliant or non-compliant. This is a small step to start with.

Second, developers assume the licence document to be too legal and tend not to read it. But remember, all the initial open source licence documents were formulated by the technical people and not lawyers. "Just take the responsibility to go through the documentation as it may be too technical for lawyers," Avci adds.

Mukundaraj says that lawyers help to enhance protection. "As developers, read documentation with a technical perspective. If you feel it does not satisfy your project, then go to a lawyer. If you feel unsatisfied, push it through

scanning tools; approaching a lawyer with the list shouldn't be difficult after that. If you are a large enterprise, you will have your own legal team and if you are a small group, look for support from incubation companies," he says.

He explains that it is still fine to trust the instincts of the technical team head: "If your technical head says that it's fine to be used as per my knowledge, go ahead and build your software. Before delivery, just ensure the scan is done, and list all the licences and copyrights. Later, you may seek legal advice to see which licences can help you achieve the compliances."

Compliance is a multi-step process. Apart from the version upgrades, your old version of software may also have some new vulnerability, which is found later. All the vulnerabilities cannot be found in the initial release days of a version of a component. Some security risks pop up after use. Hence, it's important to either follow various kinds of checks manually depending on your product line or find these automatically using tools. This helps to fix security risks immediately and to keep a constant check on licences.

A way forward

There are tools available that reveal the components used in a product, but if there is lack of understanding about the licence implications, there can be a potential risk in using them. Companies that have an interest in using standardised open source can take the help of global projects like OpenChain, which is hosted by the Linux Foundation. "It tells you how to manage the supply chain of software and, more importantly, how to prove to your customers that there is no risk in using the software. It's an ISO standard now; you can get a certification that says you have got the processes that manage open source software development. This gives a lot of confidence to customers," Callinan explains.

Well-known litigations with respect to open source compliance

- **CoKinetic vs Panasonic Avionics:** CoKinetic and Panasonic Avionics were both developing Linux based flight entertainment systems. Panasonic incorporated a massive amount of open source modules, programs, and libraries into its Linux based core software without distributing the source code. CoKinetic claimed that Panasonic was following monopolistic practices and had intentionally violated the GPLv2 open source licensing requirements, preventing its competitors from being able to build the software for in-flight entertainment services.
- **Software Freedom Conservancy vs Vizio:** Software Freedom Conservancy (SFC) has filed a lawsuit against California TV manufacturer Vizio Inc. alleging repeated failures to fulfill the requirements of the General Public License (GPL). In its complaint, SFC alleges that Vizio's TV products, built on its SmartCast system, contain software from the Linux kernel and several open source programs. It further states the sale of the TVs constitutes unfair distribution of the GPL licensed software leading to breach of obligations. SFC claims the source code for the SmartCast system must be made available by Vizio. This case highlights the importance of understanding the requirements and risks of using open source software.

According to experts, OpenChain ceates global transparency and uniformity with respect to open source compliance. It helps organisations to customise and developers to comply with various licences.

"As the group lead for the open source education board at OpenChain, we are developing an open chain compliant or awareness programme that will be available as Web based training. We are planning to host it in the Linux Foundation free of cost. This certification will help achieve awareness on the minimum criteria for compliance. We also have some ideas about creating a FAQ kind of playbook for open source compliance. Currently, a playbook for medium scale companies is available and more upgrades will be done. The course titled 'LFC193 - Introduction to Open Source License Compliance Management' is now available and the advanced version of LFC194 will be launched soon. Developers just have to continue participating and contributing to all the OpenChain education or any tooling meetings," says Mukundaraj.

Callinan warns that stricter regulations are forcing tech companies to submit their SBOM. The US White House has imposed these regulations to improve the nation's cyber security, and this can affect all software exported to that country. "There will be a point where software solutions cannot be exported to Europe or the US, unless they comply by these regulations. So, start complying with them right away," he concludes. END

The article has been compiled from a panel discussion held at Open Source India 2021 (virtual). The panelists were Biju K. Nair, founding partner, Legalitech (moderator); Aykut Avci, global license compliance manager, iText Software; Martin Callinan, director, Source Code Control India Pvt Ltd; Balakrishna Mukundaraj, associate project manager, Bosch Global Software Technologies Pvt Ltd; and Sriharsha Narayanam, lead architect, Huawei Technologies India Pvt Ltd.

Transcribed and curated by Abbinaya Kuzhanthaivel

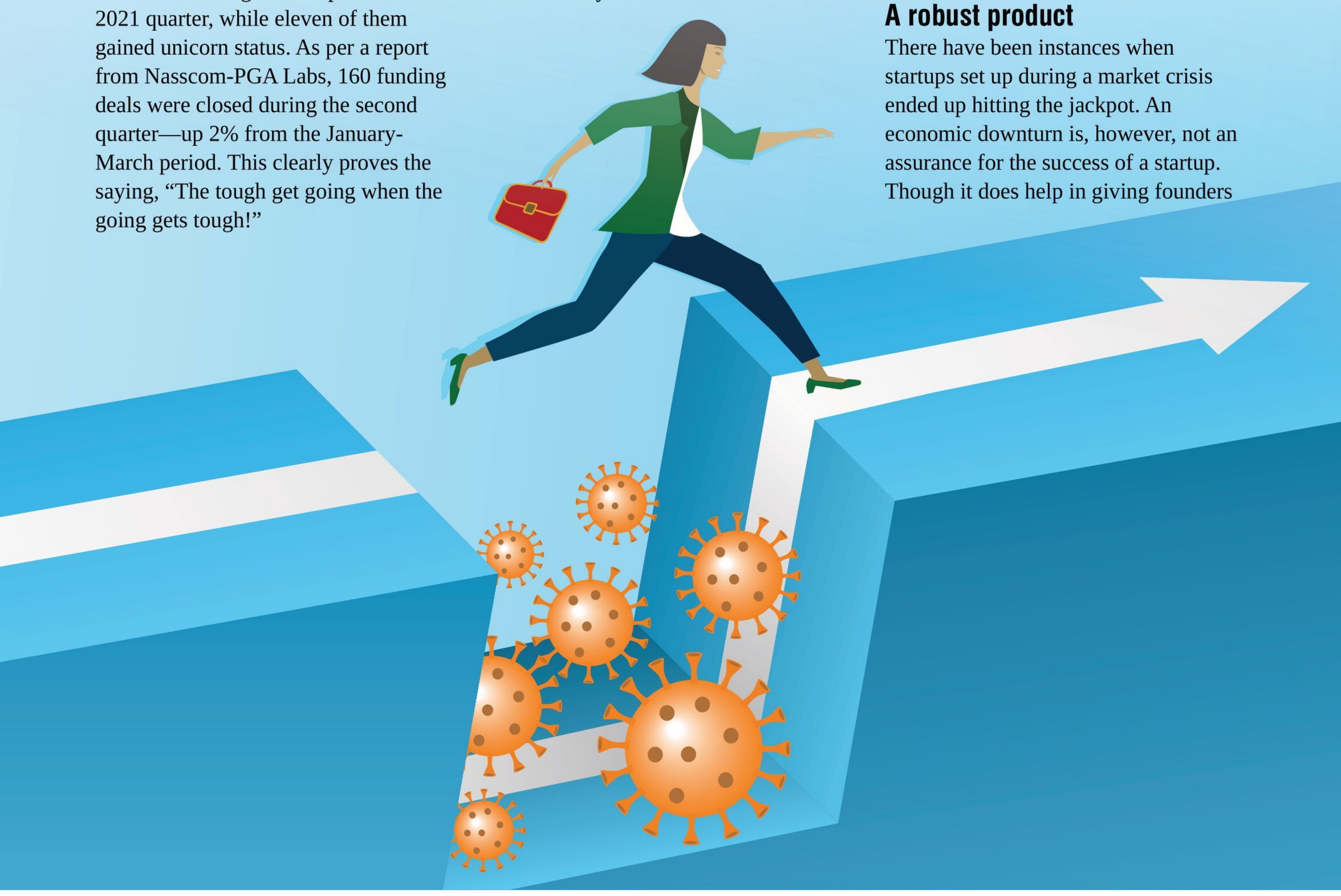
Pandemic Pains: Building A Startup in the Post-Covid World

The Covid-19 pandemic has upset the world in unimaginable ways, making the norms of business turn topsy-turvy. Now the question arises, is it a good time to take the plunge and build your own startup? Has setting up a startup changed with the advent of the 'new normal?'.

Starting a company from scratch is not as easy as it sounds. Building one in the middle of a raging pandemic could be even more difficult. But some facts appear to be quite opposite!

Indian startups bagged about $6.5 billion in funding in the April-June 2021 quarter, while eleven of them gained unicorn status. As per a report from Nasscom-PGA Labs, 160 funding deals were closed during the second quarter—up 2% from the January-March period. This clearly proves the saying, "The tough get going when the going gets tough!"

"I don't think there is a bad time to start. There are many successful companies that started during a financial crisis when there was virtually no funding available and the entire market economy was collapsing," says Suresh Jayaraju, founder and CEO, Nova Varsity.



However, there have been some subtle and some not-so-subtle changes in market dynamics, which are expected to continue evolving as the world slowly settles down. And clearly, the process of building a startup needs to evolve too.

A robust product

There have been instances when startups set up during a market crisis ended up hitting the jackpot. An economic downturn is, however, not an assurance for the success of a startup. Though it does help in giving founders

a tunnel vision where they focus solely on the quality of the product and how it can solve the customers' problem.

Ramashish Ray, managing director and CEO, Intartic, says, "This kind of situation provides a lot of opportunities as it provides less of an entry barrier, since many of your bigger competitors are busy solving other problems."

The start of a venture still requires a lot of work that needs to be done. Jayaraju advises, "I think it is a great period to be focused on inwardly developing your product, or your solution matrix, and actually testing it out. This sort of gives you a cocoon where you can stay away from market dynamics or distractions and be absolutely focused."

Self-validation

Building a startup takes enormous amount of time, energy, and resources, which is why the process of self-validating the seemingly remarkable idea is crucial. "The purpose of self-validation is to come to a decision of should I go ahead or not," says Jayaraju.

Self-validating your idea before fully investing in your startup can help you make the right choices. During self-validation, the idea may not seem fit to be taken forward. This could be due to bigger competitors, the economics which do not seem to work, or the product suits only a niche market. Proper analysis could compel you to not to go ahead with your idea or change it to improve its chances.

Validation of the product needs to come from the end-users rather than just family members or close friends. Various mentors, tools, and techniques are available to validate a product but, ultimately, the real test is whether the customer is willing to pay for your product.

Figuring out how well your idea, product, or solution fits the overall market demand is a crucial step, called the product-market fit. As per Jayaraju, "Product-market fit is very simple. The user of your product, whether it is a business or the end-consumer, should start missing your product on a daily basis. That is when you can say that you are getting a huge demand which is unlimited."

Vijetha Shastry, executive director, TiE Bangalore adds, "You should have the ability and gumption to create something on your own rather than just relying on receiving funds. If you get twenty customers to pay you in advance, you create a product of resources."

For self-validation, Jayaraju suggests the simple method of showing the product to at least fifty people from various industries/demographics and taking their feedback. He insists that keeping this as a fairly sacrosanct number is important, particularly because the kind of response you may get from just five people could be completely different from what you might receive from fifty people. He says, "Confusion arises if the number is less, as some people may or may not like your idea. But the moment you move to fifty, you will have an accumulation of so many views. The aggregation of those views helps you know what points to pick up and what not to."

Once you have gathered a clearer picture of what may work for the people, it is time to create a prototype, or a minimum viable product (MVP). The prototype is not the final product, and at best can be considered as a preliminary version of what it could be. It is advised that you spend minimum time and resources on building it.

Jayaraju comments, "One of the biggest mistakes that startups end up making is that they fall in love with the product and not the solution and spend months in building the product. Instead, I think they should be very scrappy about building the first version and ideally aim to finish it within a maximum of thirty days. You can build your second, third, or fourth version based on the users' advice, and give it to them for free, even if you think your product is going to be highly-priced, so that they can try it and provide their feedback. This is a really important step after self-validation."

This helps you track which version of your product has the maximum potential. The feedback given for the product is valuable as it helps you understand what a customer finds useful and what aspects of the product need to be changed. According to Jayaraju, this is a foolproof way to understand what value your final product provides to the customer.

Finding the ideal team

The people involved in starting a startup can also be called the founding team. It is critical to choose the right people in the team who understand your vision for the company and the product. No one person has all the skills required to build a startup, which is why choosing people with varied skill-sets helps to a great extent.

But that's not all. It is also important that the founding team members have the right chemistry with each other. There needs to be a clear understanding between the founders of their competencies and their role in building the startup.

If you are working full-time as an employee in another company, there is no way you can put in your full effort into the startup. In such a case, a team can help share efforts and time. Jayaraju opines, "Cultural mingling of teams and how they get along is very important. The willingness to sweat it out and not sleep is the sort of passion that breeds a very different level of startups."

Competencies and skill-sets to look for in a team can vary based on the vertical under which the product functions. For a tech startup, it is important to have someone with a broad technical background and someone with a broad business development background.

A Unique Idea: Does It Exist?

Many times, founders are dumbstruck when their idea seems to have been already put into practice by someone else; and this can be quite disheartening for them. However, there will always be similar ideas, and it is very rare that an idea occurs that is so very different from the others'. But if you can solve a problem for your customers better than the others, there's nothing that can beat you. And there is nothing like good execution.

The chief technical officer should be one who understands the mechanics of the product being built. Having a group or an expert for business development is equally important for figuring out a suitable business model and scaling the startup to next level. Overall, it can be said that stronger the foundation, the higher the fortress goes.

The first 500 days

The initial days of a startup are a deciding factor of how its journey would look like. With much to figure out while building a company, the learning really has no end—no matter how big it gets.

Notably, the initial days pass by in figuring out and having a stable business model in place. You try to have a steady cash flow from a stable customer base and try to keep your input and output in sync. Ray says, "A clear balance sheet should start emerging because 500 days is more than a year and, by that time, you should have a balance sheet that speaks for itself. It's not the scale of the balance sheet, or how large the balance sheet is, but how robust it is."

By this time, the founding team members must also be clear about their role and where they should aim to take the company. Ticking off boxes of some of the goals that a startup should hit in a stipulated time can provide a clearer and better picture of how the future would look like for the company. While not achieving them doesn't mean the startup is doomed, it shows how well the benchmark is set.

Ray adds, "There must be some predictability that if you do this particular task then the probability of hitting this goal is high. A lot of people talk about success, but if you do not have the right parameters, then it is difficult. You need to be very clear about what you're looking for because if you miss that, the test board itself is wrong and you will keep thinking you are going the right way even when you aren't."

Ray notes that after the first 500 days, a startup should be essentially ready to scale up. But to reach that stage, it is important to have a sufficient cash flow that will help it survive the initial 500 days.

However, scaling up a startup is easier said than done. A startup usually begins and survives its initial stage through bootstrapping and the founders' own capital. Scaling up usually requires funds on a much larger scale. "The game of scaling up is very different from the game of starting up and proving your product's worth," notes Ray. Here is where the knowledge of a mentor or incubator comes in handy.

Accelerators and incubators

Investors, venture capitalists (VCs), accelerators, and incubators play a significant and influential role in startup ecosystem. They are the backbone of startups, providing a lot of grounding and mentoring to get access to the market.

Investors have an undoubtedly wide network, definitely much wider than any budding entrepreneur can boast of. They get exposed to various VCs, embassies, and government projects that gives them access to awards, funding, and grants, and opportunities to showcase the product. Their access to various trade commissions and embassies helps in showcasing startup projects in those countries and enter those geographies.

Creating a perfect proposal for investors can be tricky, but it need not be. Shastry wants the startups to remember the investors, "If they see that the team is strong and the credentials of the team are relevant to what is being solved, then that is something that they definitely look for. The moment the investors see that both the team and the ability for execution and pivoting towards the problem solution is strong, they fall in love with such types of startups."

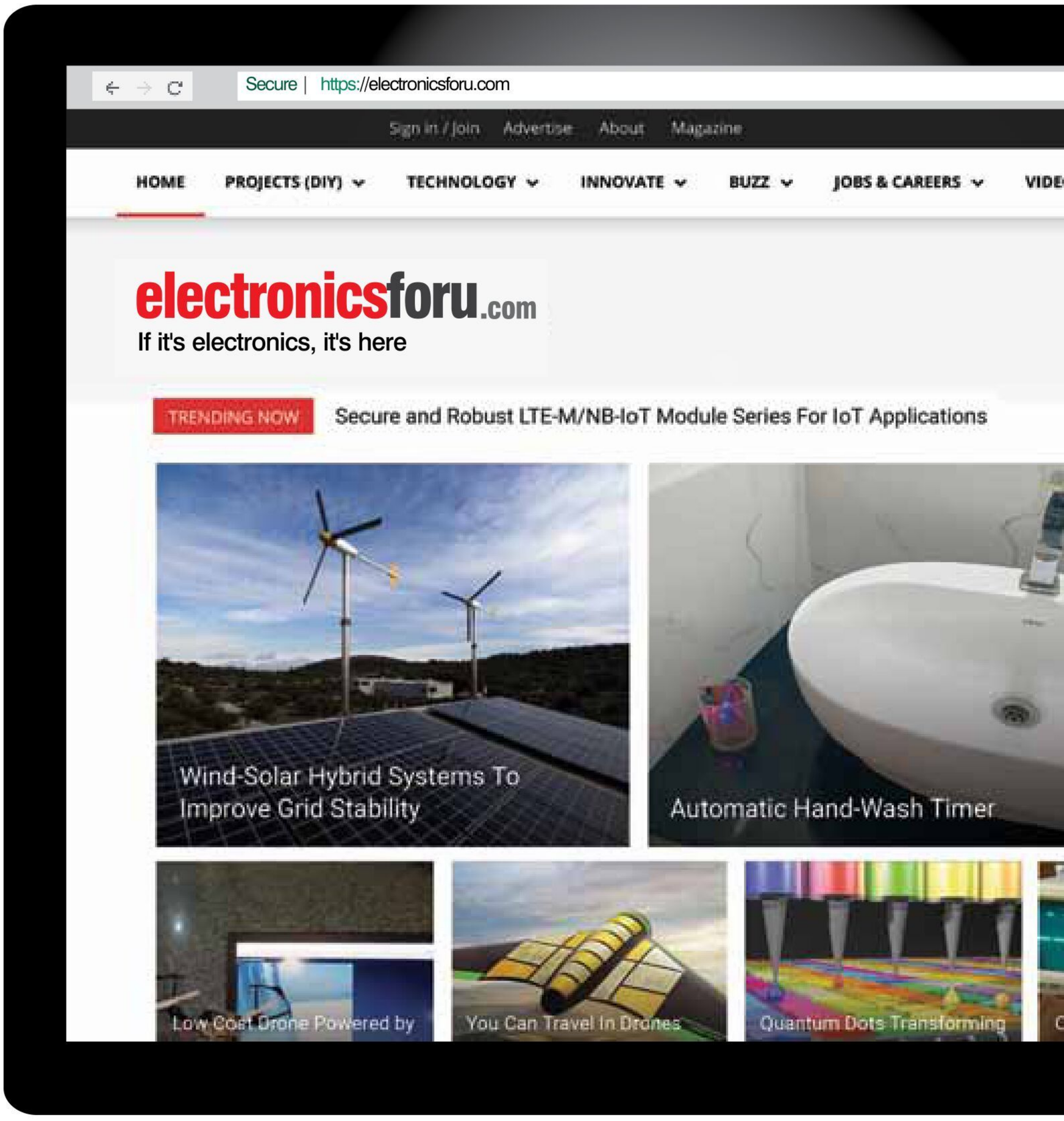
In spite of all this, there are very few startups that actually end up getting funds from VCs and incubators, and the competition has become even more rigid post-Covid. However, there is no reason to be disheartened if an investor rejects you. At no point in time does getting into an incubator define your success or failure.

Jayaraju, in response to the situation above, reassures, "I think the best thing about VCs is that the more they reject you, the more you tend to refine your proposition and make your business model even more successful. You never know if you're gonna be successful after the first pitch or the sixtieth pitch, but by asking the right kind of questions, you can still add value. Even if they don't fund you, most of them are open to giving you suggestions, and they are more than happy to help you connect with other people,"

In support, Ray concludes, "Sometimes a startup journey can be a very lonely one and may seem like you're just sitting as a frog in a well. It is the incubators, accelerators, and VCs

Continued to page... 31

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

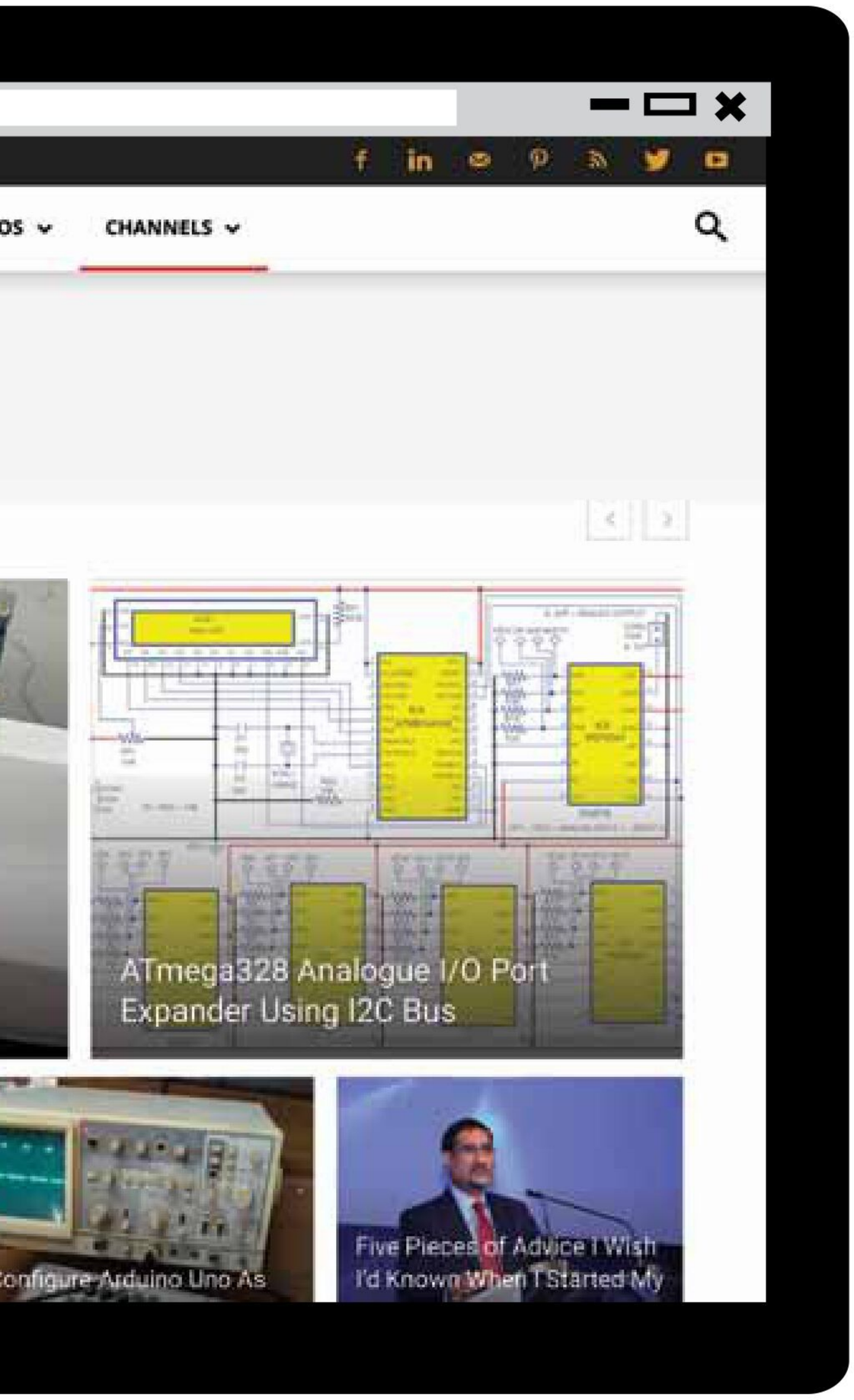
READERS

- 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...





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.

Amazing DIY Projects. Latest Tech trends.

The hang-out for electronics enthusiasts.

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

Greenfield Technology: How Open Source Software can be Useful

Organisations can today use open source software to scale applications and platforms with true agility. There is a range of open source software available to build an organisation from the ground up in a greenfield project.



Greenfield implementation refers to building an organisation from scratch, without modernisation or transformation of legacy platforms. Organisations of all sizes are seriously considering greenfield projects after disappointing attempts at brownfield technology implementations that focus on advancing the existing technology landscape little by little. Limited success in modernisation of existing platforms or transformation initiatives has given an impetus to greenfield technology.

Design and architecture play a key role in greenfield implementation. Even though there is no baggage of legacy technology or dated processes holding one back, a new path ahead poses its own risks. To address the concerns head on, one needs to set an overall technology strategy, and define what is fit-for-purpose and fit-for-use across each business function of the organisation to enable the right selection of open source software and platforms.

We know that open source software (OSS) is software that's released with source code and is licensed, with copyright holders granting the rights to use, change and distribute it to anyone for any purpose. But things are not so simple! Commercial software providers enter the OSS space by releasing early versions of their software for free so that open source enthusiasts can improve

PIONEERS IN TECHNOLOGY.
PARTNERS IN SUCCESS.

EFY GROUP
YOURS SINCE 1969





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

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

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

D-87/1, Okhla Industrial Area, Phase 1, New Delhi-20 • Ph.: 011-26810601/02/03 • Email: info@efy.in • www.electronicsforu.com



Figure 1: Typical technology functions of an organisation in supporting business

it. They keep minimal functioning software for free and market it heavily. They end up confusing developers and organisations, who think they have adopted OSS based application development and give into the ease-of-use hype, but end up paying large sums of licensing fees down the road. Rewriting the application to get out of using the software takes time and money, which most organisations don't have when they are growing. This is exactly why there should be a focus on design and architecture while selecting the right open source platform, software, and tools in developing applications from the ground up.

Business functions of an organisation

The first step in developing technology strategy is to understand the business functions of the organisation. Once the functions are broken down, the listing and consolidation of the core technical capabilities of competitors will provide a holistic view of what's needed to get a tech organisation up and running.

A typical organisation consists of many business functions; however, many of these do not need software development. One needs to evaluate if a business function will act as a

Business function	Core capability	Open source software(s)
E-commerce and mobile app	Online shopping portal on the Web and on mobile devices working across multiple popular browsers and mobile operating systems	Open Cart, WooCommerce, Zen Cart, Shuup, Ubercart, Spree Commerce. PhoneGap, Flutter, Ionic, BuildFire
Customer relationship management	Managing customer information and interactions across all touch points	SuiteCRM, openCRX, EspoCRM, vTiger, Zurmo
Customer service	Customer support, request tracking and management	Bugzilla, Helpy, osTicket
Marketing	Promoting and selling products and services using emails and campaigns, and generating leads with tracking	Geary, Mautic, OpenEMM, Thunderbird
Document and content management	Website and portals for internal use, customers, investors, etc; Documents management and file sharing	WordPress, Drupal, Joomla, OpenKM, 7Zip, WinSCP, FileZilla, FireFTP
Productivity tools	Creating text documents, presentations, spreadsheets	Apache OpenOffice, LibreOffice
Monitoring and dashboard reporting	Tools to collect, analyse and report data, and develop visualisation charts with dashboards and notifications	Graphana, Datawrapper, Charted, Draw.io, D3
Accounting and finance	Managing resources and assets, order and invoice management, compliance reporting	Apache OFBiz, ERPNext, Tryton, GNU Cash, OpenBravo, GanttProject
Operating system (OS) and networking	Platform to run applications and connect across applications, tools and end-point devices	Linux, Ubuntu, Nagios

Table 1

core service offering or is part of an administrative function. As an example, HR and payroll functions can be outsourced from a technology perspective for significantly lower cost as compared to developing in-house software applications.

Nine blocks of core business functions have been identified where typical organisations like to have all or some level of in-house application development. Table 1 looks at each business function for its core capability and the OSS options for it.

Most organisations have some exposure to open source software, including that which is sold commercially. Today, there is a range of open source software to choose from to build every desired business function of the organisation. As long as one is willing to take the time in understanding the requirements, there is some open source software readily available to get started with greenfield technology implementation. END

References

[1] Open source organisations, *https://opensource.com/resources/organizations*
[2] Apache community led development, *https://www.apache.org/foundation/*
[3] Open source cheat sheets, *https://opensource.com/downloads/cheat-sheets*
[4] Mobile app development with Flutter, *https://flutter.dev/*

By: Bala Kalavala

The author is a technical architect, evangelist, thought leader and a keynote speaker. He currently works as a distinguished member of the technical staff and heads the enterprise architecture practice as chief architect at Wipro Ltd. This article expresses his views and not that of Wipro.

Continued from page... 25

who ensure that you don't remain in that well but also go out and see what is going on around the world, because they are meeting multiple people like you across the industry. They can give you that reality check and sometimes be the nudge that puts you on the right path."

Apart from that, they also look into things like validation core, crowd infrastructure, as well as technology infrastructure. It's not possible for a hardware startup to have huge infrastructure right away, which is why being part of an incubator is so valuable.

The bad and the ugly

Creating anything of your own is as exhausting as it is exhilarating. Every successful startup is the result of the sweat, blood, and tears poured into it by its founders. As inspirational as it sounds, sometimes a lack of cushioning, either financially or emotionally, can do more harm than good. This is more so in present times where startups are evolving every day and it is getting increasingly difficult to set yourself apart.

"A lot of entrepreneurs and founders are exploiting themselves. They are working without taking a break, not being paid salaries, not taking care of themselves nor the people around them. What they're doing is they're just killing themselves, which they shouldn't," notes Shastry.

Even so, Jayaraju argues that too much cushioning can also have an adverse effect such as complacency. "You do not need to be self-exploitive but at the same time, unless there are sacrifices made or there is sole devotion to a certain cause, you may not end up making great products or solutions, because you have a backup. So, the answer really lies in questioning yourself: How much risk are you willing to take?"

The formula for creating a successful startup has not changed much post-pandemic. However, the importance of building a robust product or solution as a tech startup has never been made clearer. With the future looking bright for startups, the founders need to find the middle ground between their passion, customers, and what would make money.

In conclusion, Ray states, "The startup founder is 'a buck stops with me' journey. You and the founding team are alone, and once you let that seep into your flesh and blood with added thick skin, you are mentally ready whatever the time may be." END

The article is based on the panel discussion 'Idea To Startup: How To Do It?' presented at September edition of Tech World Congress 2020.

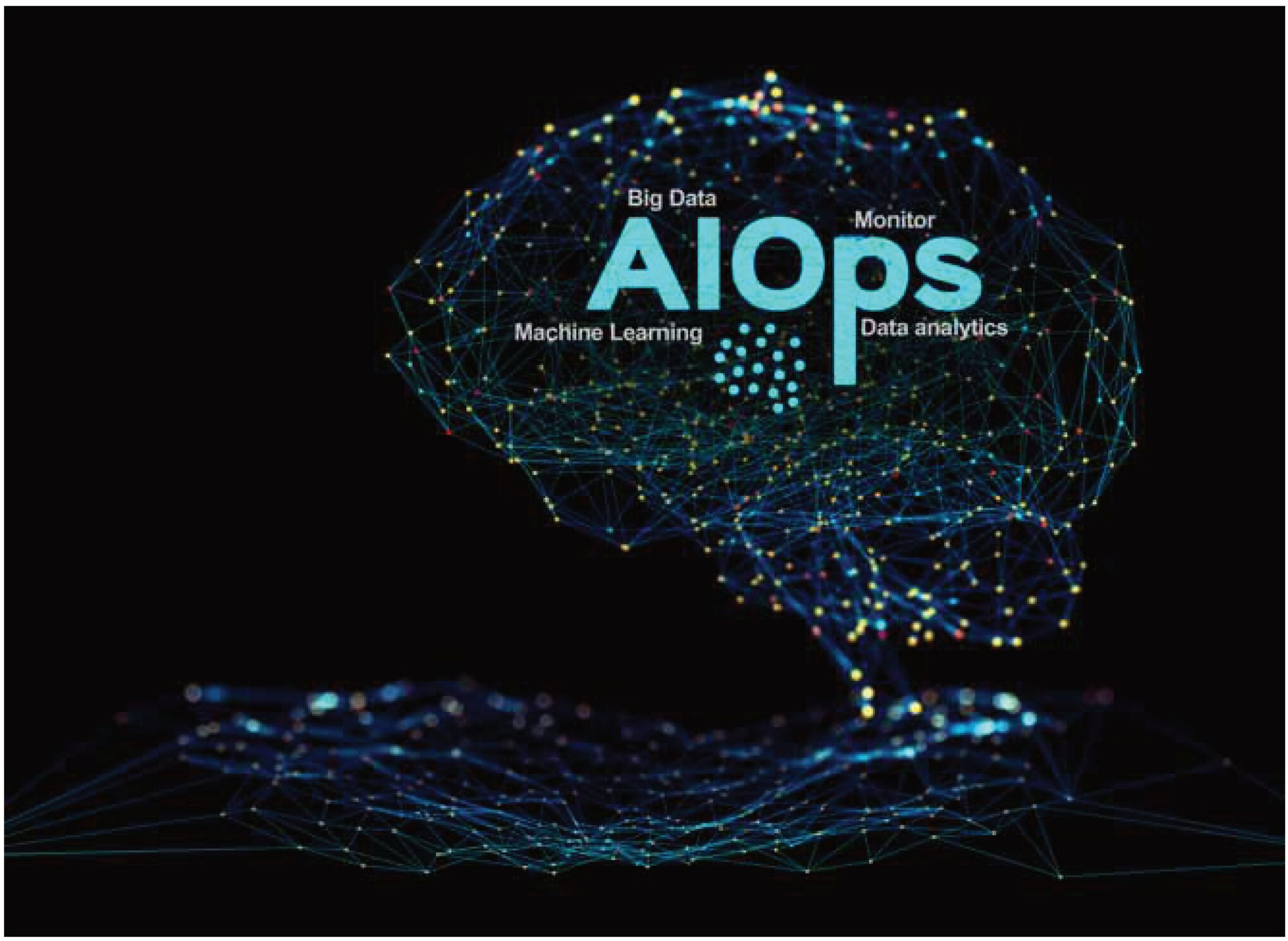
By: Siddha Dhar

The author is a business journalist at *EFY*.

The article was originally published in the December 2021 issue of Electronics For You.

Why AIOps is Essential for Businesses Today

Digital transformation is constantly forcing many enterprises to move towards automation. IT leaders are now turning to artificial intelligence for IT operations (AIOps) to enhance their businesses, improve risk management, deliver faster services, and get actionable reports for better analysis.



Artificial intelligence for IT operations (AIOps) helps businesses become more efficient. It intelligently assists enterprises by defining actual issues using predictive analytics. It also focuses on the most critical issues that need to be resolved, getting to the root cause of problems, and offers accurate recommendations to correct them while reducing MTTR (mean time to resolution).

What is AIOps?

According to a McKinsey report, "The most advanced deep learning techniques deploying artificial neural networks could account for as much as US$ 3.5 trillion to US$ 5.8 trillion in annual value, or 40 per cent of the value created by all analytics techniques."

IT leaders are turning to AIOps as a solution to automate mundane tasks and authorise their IT teams to deliver better services and get actionable insights. In simple words, AIOps is a solution that includes machine learning

EFY GROUP
YOURS SINCE 1969

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



SUBSCRIBE AND **SAVE**

ORDER FORM

PRINT MAGAZINE	1 YEAR (12 copies each)	3 YEARS (36 copies each)	5 YEARS (60 copies each)
		WITHIN INDIA (IN RUPEES)	
Electronics For You (Rs 100/copy)	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.

and data analytics into IT operations to troubleshoot, monitor, and achieve other network management goals.

What challenges does AIOps address?

To stay ahead in this competitive era, businesses must ensure the end user experience remains flawless and meets continuously growing demands. Technology has become important for most businesses, but it also comes with major challenges.

IT has traditionally worked in silos. Each component of the technology stack is divided among teams including developers, network admins, security admins, and business leaders. They are completely unaware of the things that are happening around the infrastructure. AIOps can help enterprises break down processes and predict failures. Let us go through some of the challenges that enterprises can address using AIOps.

Problem resolution: AIOps is the best solution to reduce silos and reshape an environment that can provide comprehensive monitoring across all aspects of the IT infrastructure. AIOps enabled automation is important for identifying and resolving problems in real-time as soon as they come up and before the end user is impacted. The automation can save time for IT teams so that they can focus on other essential tasks.

Identifying the root cause: IT teams need to quickly identify the root cause of an issue. Enterprises spend a massive amount of money on their IT infrastructure, so they need to focus on how they can monitor their events, logs, errors, and anomalies on a centralised AIOps platform that can quickly compare all these metrics, point out issues, and automate remediation, thus reducing potential outages.

Making space for innovation: With an AIOps strategy and framework in place, most mundane tasks can be eliminated. With automated processes, enterprises can get insights into what precisely happened and what was done to fix an issue, as well as what was impacted during the process. This frees up time for teams to focus on innovation, helping them reduce IT costs and eliminate the risk of losing customers.

How and why are enterprises using AIOps?

Before implementing AIOps, enterprises should understand its concept and capabilities, and learn how it can be used to regulate IT operations. AIOps can be applied and enhanced from time to time, as it is capable of learning things very quickly. Let us see how and why enterprises are using AIOps.

Instant alerts: Enterprises use AIOps for delivering intelligent alert notifications that let IT teams understand event history, resolve incidents, and meet service-level needs for faster problem resolution.

Root cause analysis: Organisations want to ensure better uptime and steadiness of services. Root cause analysis with a quick diagnosis of problems helps them to predict the impact as well as regulate the cause and effect of operational issues.

Threat detection: Enterprises leverage machine learning algorithms to quickly identify threats through pattern recognition. AIOps allows IT teams to extract signals from noise and recognise events that vary from regular system behaviour.

Visualisation of incidents: AIOps platforms are helpful to organisations in highlighting incidents that can cause problems and need immediate attention. Relevant performance metrics can support incident responders to get quick resolutions to various issues.

The core aim is to reduce incident and ticket volumes, significantly reduce the time teams spend on the first response, prioritise the alerts, and find the root cause of the incident. Enterprises use AIOps to eliminate low-value and repetitive tasks as well as increase the potential of the entire infrastructure, and for quick issue remediation through faster root cause analysis.

Does your business need an AIOps solution?

AIOps tools are already getting popular. This indicates a shift towards a modern IT strategy — from using machine intelligence to getting enhanced support for digital transformation. Many of you must be wondering if your organisation needs AIOps or not. Here are some of the key questions that will help you evaluate the right AIOps tool that meets your business requirements:

- What problem will an AIOps solution resolve?
- What are the key features of the AIOps platform that your enterprise will need?
- How will advanced analytics help IT operations?
- What kind of benefits can AIOps bring to your business?
- How do AIOps tools differ from normal network monitoring tools?

Once you have the answers to all these questions, you will get a clear picture of whether you need an AIOps solution or not. If you need one, you can look for various AIOps solutions that are available in the market, compare them, try their free trials, explore as much as possible, and then make the right decision. END

By: Amit Shingala

The author is the Co-Founder & CEO of Motadata (Mindarray Systems Pvt Ltd). He holds over 13+ years of experience in building, scaling & selling enterprise products to solve complex problems of cross-monitoring by gathering meaningful insights from a hybrid infrastructure by leveraging data ingestion from heterogeneous sources and ML capabilities.

Designing Java Microservices with Spring Boot



Image source: *http://www.freepik.com*

Microservices are polyglot. Developers choose different platforms for building different microservices, as appropriate. Java is still the top choice for developing microservices for enterprises and Spring Boot is the *numero uno* framework for the Java platform. In this sixth part of 'The Design Odyssey' series, we will explore the design aspects of Java microservices by building *AddService* of UMS using the Spring Boot framework.

Once upon a time, software was built and deployed as a single monolith unit. It was difficult to maintain such a monolith in terms of adding and removing features, and fixing the defects. Over a period of time, modular systems replaced monolith systems. However, the modularity was maintained only during the development and not the deployment of software.

For instance, a team of Java developers builds a large e-commerce application as a set of multiple modules like order management, catalogue management, etc. But, the whole code is built together and packaged as a single WAR file. Then the team deploys the WAR file into a Web server like Apache Tomcat. As shown in Figure 1, the monolith WAR needs to be rebuilt and redeployed as a whole, for every change. Because of this, the end user experiences frequent disruptions whenever the team reboots the server for rolling out new features. To avoid this, teams usually delay the feature rollout for weeks and months. Such a delay obviously results in business loss.

The root cause for all this lies in deploying the application as a single monolith WAR. What is the solution?

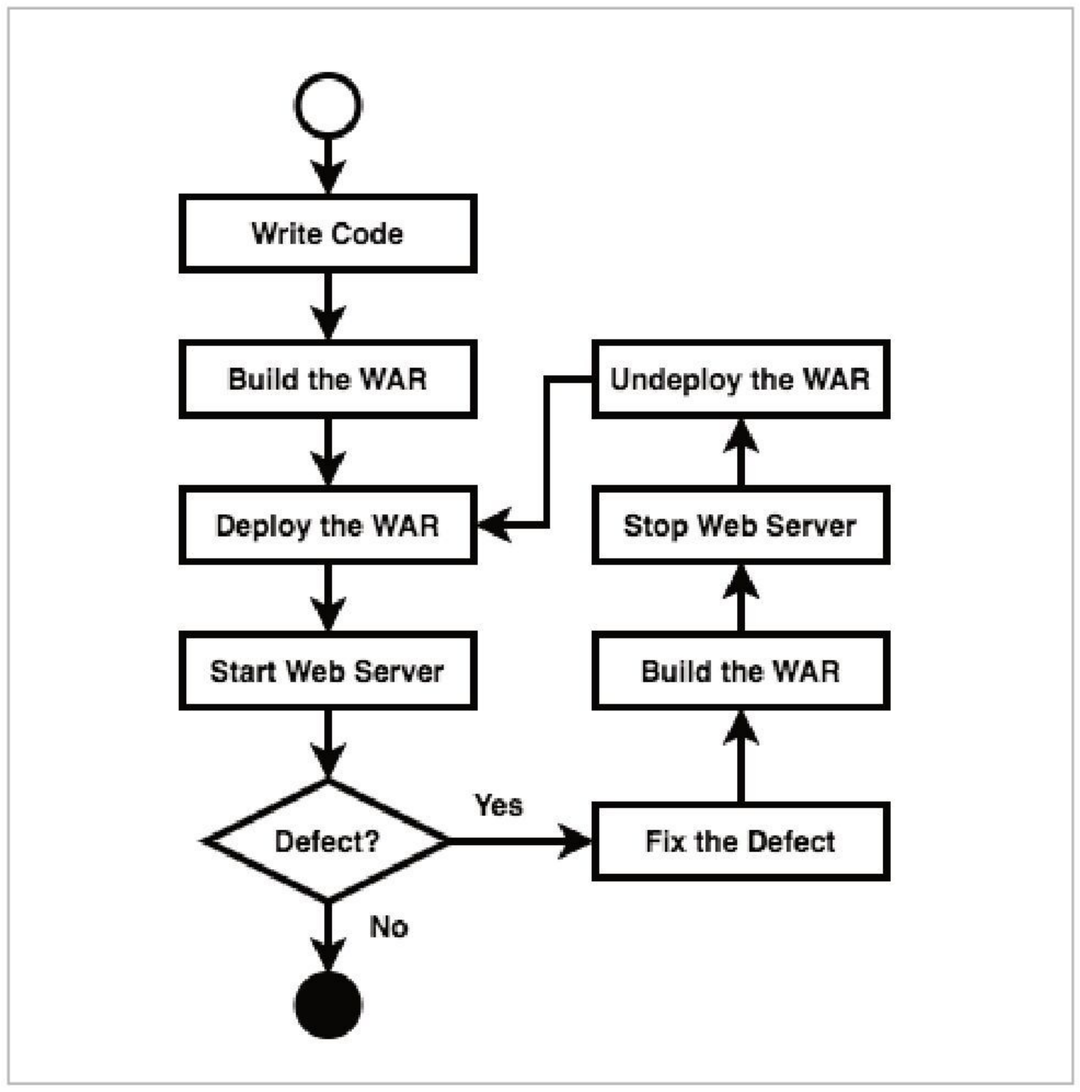


Figure 1: Traditional development and deployment

Microservices

As indicated in the previous part in this series, the solution to the above problem lies in the microservices architecture. If the modules of order management, catalogue management, etc, are not only coded independently but also built and deployed independently as microservices, the problems cited above will automatically be resolved. This approach allows the team to build and deploy a given microservice any number of times, as frequently as it wishes. The end user will never experience a total shutdown of the e-commerce application.

An EFY GROUP EVENT

5243
ATTENDEES

70+
SESSIONS HOSTED

100+
SPEAKERS

18
COUNTRIES REPRESENTED



18th Edition

OPEN SOURCE INDIA

VIRTUAL EVENT

Thank You

VISITORS, for making the event a huge success

SPEAKERS, for contributing your valuable thoughts

PARTNERS, for your support

Microsoft Azure | soda foundation | Google | aws

MySQL | HUAWEI | iTEXT | GitHub

SUSE | ENTREPRENEUR FIRST JOINEF.COM | TRUEFOUNDRY BY ENSEMBLE LABS | Application Library Engineering Group

openLooKeng | EDGE GALLERY | soda foundation meetupindia | BANGALORE CNCF MEETUP

www.OpenSourceIndia.in

For more details, call on +91-9667743666 or email us at info@opensourceindia.in

Java microservices

The enterprise market has been traditionally dominated by Web architecture. The Java platform has been the automatic choice for developing enterprise Web applications because of the proven servlet technology, along with numerous frameworks ranging from the old Struts and JSF to the current day Spring. Figure 2 shows the architecture of a traditional Java Web application deployment.

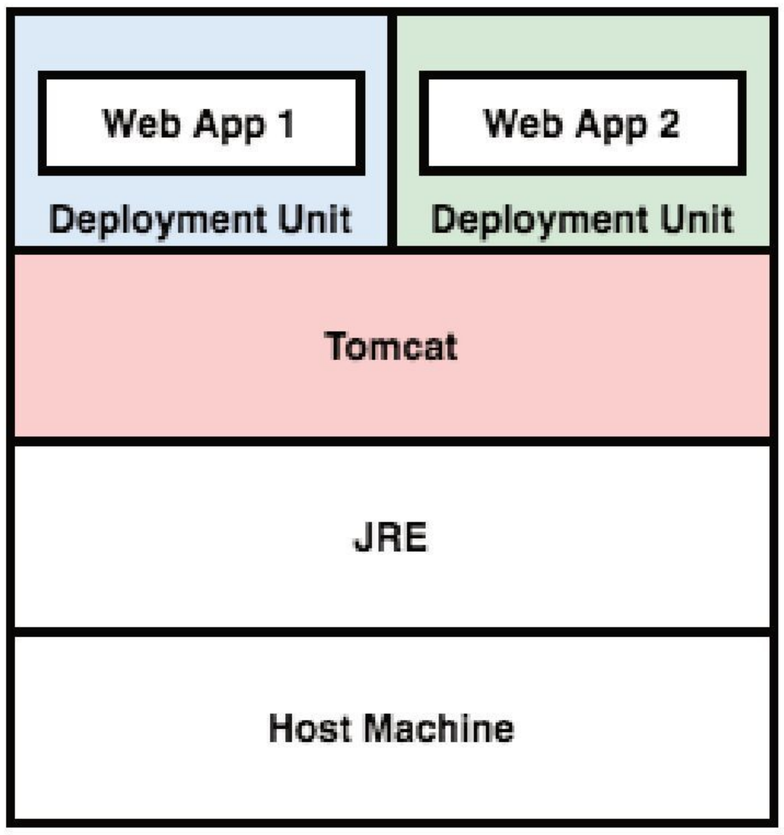


Figure 2: Deployment stack of traditional Java Web application

In the traditional approach, the Tomcat Web server had to be installed and provisioned before deploying the Web application. Moreover, multiple Web applications used to be deployed on a single Tomcat Web server. In other words, the deployment unit consists of just the application code. This approach is not compatible with the microservices architecture. Why?

Since microservices are expected to be deployed frequently (several times a day in some cases), the deployment process should be short, automatic, and smooth! This implies that the process of pre-installing and provisioning the Web server is clearly not an option anymore. On the other hand, as the microservices are independent of each other, co-hosting multiple services on the same Web server is also ruled out. Then, what is the way out?

Spring Boot framework

There are several frameworks available for building microservices on the Java platform including DropWizard and Vert.x. However, Spring Boot is the most popular framework in the community. It is built on top of the time-tested Spring framework and comes with all of the latter's proven features like dependency injection (IoC), aspect injection (AOP), Web context (MVC), JPA integration (ORM), etc. Naturally, it is very easy to adopt Spring Boot if you have worked with Spring earlier.

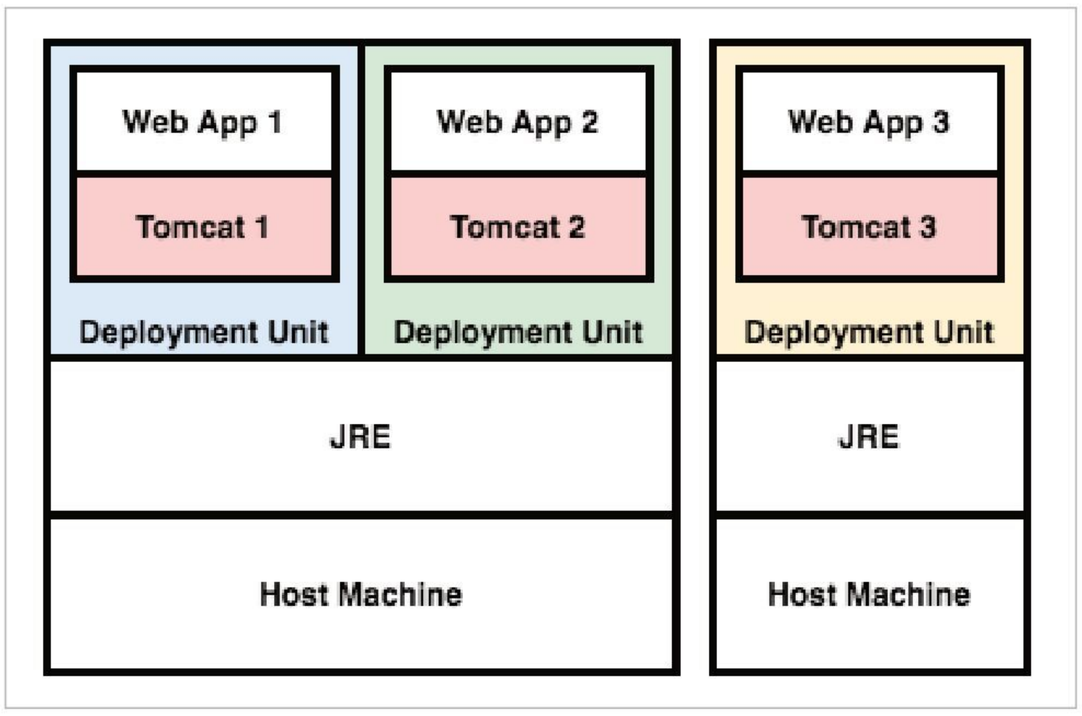


Figure 3: Deployment stack of the Spring Boot application

So, what is the value-add that Spring Boot offers? Spring Boot is an opinionated framework that prefers convention over configuration or coding. In other words, it simplifies the job of the developer.

It also comes with an extension to the Maven. This extension packages a microservice as a JAR, as opposed to a WAR file. Not only that, the JAR file even consists of an embedded Tomcat Web server. What does this mean? As long as you have a JRE, you can simply place the JAR file in the *classpath* and start the service. Spring Boot takes care of starting the Web server and mapping it to the service endpoints.

In other words, with Spring Boot, you no longer pre-install the Tomcat, no longer pre-provision the Tomcat and no longer co-host several applications in the same Tomcat. Instead, each application is self-sufficient with its own Web server, as shown in Figure 3. To summarise, the deployment unit consists of application code along with all the required libraries and Tomcat. Naturally, you are free to deploy the applications on the same machine or on multiple machines without worrying about setting up the server beforehand.

Let us build a Spring Boot microservice to demonstrate this.

UMS *AddService*

In the previous article in this series, we have developed a design for *AddService* as part of UMS. That design proposed a domain model, application layer, and infrastructure layer on the lines of the Onion model of domain-driven design. Though we can implement the proposed design completely on our own, why should we do it when a framework like Spring Boot offers some of those components out-of-the-box? After all, Spring and Spring Boot offer ready-to-use factories, proxy generators, weavers of responsibility chains, adapters, etc.

As you know, *AddService* offers only one functionality — to add a user to the system. In line with the contemporary style, let the *AddService* expose the functionality via REST

interface over HTTP. The following can be the REST API for adding a user:

```
POST /user
```

This API accepts the user data in the format of *application/json* in the HTTP request, and returns the newly created user data in the *application/json* format along with usual HTTP headers.

Let us do this one step at a time.

Generate the Spring Boot project

Visit the *https://start.spring.io/* and choose options, as shown in Figure 4. The following are some of the important options selected.

Project type: Maven
Language: Java
Spring Boot version: 2.7.0
Project metadata:
Group: *com.glarimy*
Artifact: *ums-add-service*
Package name: *com.glarimy.ums*
Packaging: JAR
Java version: 8
Dependencies: Spring Web, Rest Repositories, Spring Data JPA, H2 Database, Spring Boot Dev Tools, and Validation

Click on the *Generate* button and you will get the Maven project downloaded as an archive. Extract it and export the project into any IDE of your choice.

Manifest

As you know, the Maven tool uses the manifest file *pom.xml* to curate the build process. The following part is self-explanatory.

```
<parent>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-parent</artifactId>
      <version>2.7.0-SNAPSHOT</version>
      <relativePath/>
</parent>
<groupId>com.glarimy</groupId>
```

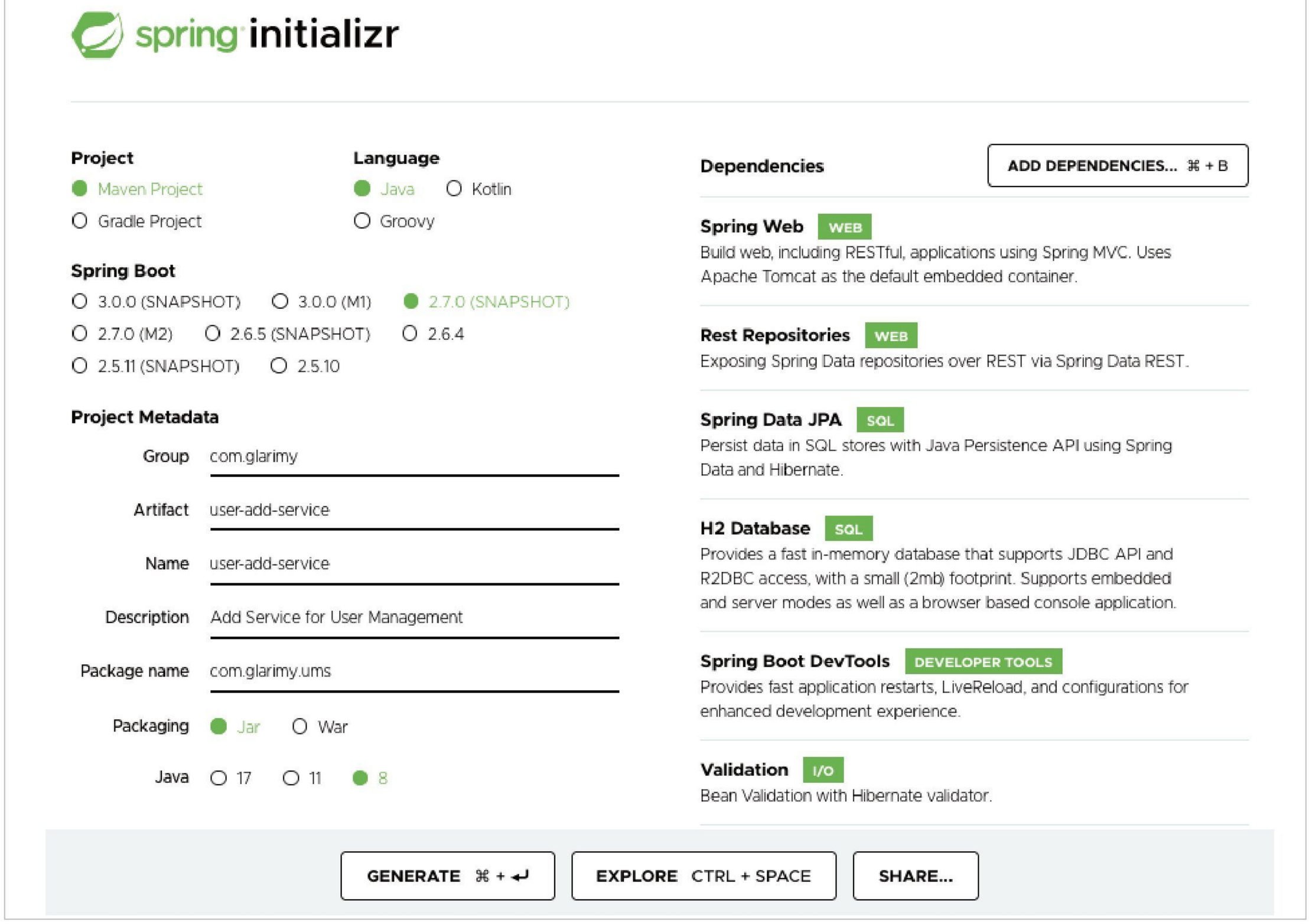


Figure 4: Spring Boot initialisation wizard

```
<artifactId>ums-add-service</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>ums-add-service</name>
<properties>
      <java.version>1.8</java.version>
</properties>
```

From this snippet, you can find that we are on the way to building an artifact named *ums-add-service* under the group *com.glarimy*. It is also clear that we prefer version 2.7.0. of Spring Boot. Different versions of Spring Boot support different versions of Java, and this one supports the time-tested version 1.8.

Now let's explore the dependencies.

The *spring-boot-starter-web* adds an embedded Tomcat Web server along with HTTP-based MVC framework and REST capabilities.

The *spring-boot-starter-data-rest* adds the capabilities to generate the repositories (recollect the Repository pattern of domain-driven design). The repositories can be backed by various storage technologies.

The *spring-boot-starter-data-jpa* adds capabilities to connect to JDBC data sources using JPA API on top of Hibernate ORM implementation. With this, Spring Boot automatically chooses JPA to build the repositories. However, the JPA must be backed by an RDBMS. The *h2* artifact adds the H2 in-memory all-Java RDBMS. You may choose any other database like MySQL to support JPA. We want to keep it simple and hence chose this embedded database server, for now!

The *spring-boot-starter-validation* adds the validation framework on top of the Hibernate Validator libraries.

The *spring-boot-start-test* adds test capabilities. The *spring-boot-devtools* adds a few useful development tools. For example, one of the tools watches the source code, and automatically rebuilds and reruns the whole service every time you edit the code.

The following build section instruments the Maven plugin that is specifically developed for Spring Boot.

```
<plugin>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-maven-plugin</artifactId>
</plugin>
```

It generates a fat JAR with compiled class files, libraries and, most importantly, the embedded Tomcat Web server.

That's all about *pom.xml*. Let us move on to the code.

The domain model

The domain model we developed in the previous part consists of the classes *Name*, *PhoneNumber*, *User*, and *UserRepository*. These fundamentally remain the same. However, we are decorating them to inject a few capabilities of Spring Boot like validation and persistence. This reduces the amount of code that we otherwise have to write.

Recollect that the *User* is modelled as an *Entity* class on the lines of domain-driven design. The *AddService* is supposed to persist the *User* objects using *UserRepository* (into H2 RDBMS, in this case). Instead of writing a lot of code for that, we decorate the *User* with JPA annotations and generate the *UserRepository* with the help of Spring Boot.

Here is the annotated *User* class:

```
@Entity(name = "ums_users")
public class User {
      @EmbeddedId
      @Valid
      @NotNull
      private Name name;
      @Embedded
      @Valid
      @NotNull
      private PhoneNumber phone;
      private Date since;
      User() {}
      ...
}
```

With the *@Entity* annotation, we are mapping the objects of the *User* to the rows of a table named *UMS_USERS*. The properties of *Name* act as the columns of the primary key because of the *@EmbeddedId* annotation. And the properties of *PhoneNumber* are included in the *UMS_USERS* table because of *@Embedded* annotation. In other words, the properties of *User*, *Name*, and *PhoneNumber* are all part of a single table named *UMS_USERS*.

The *@Valid*, *@NotNull* annotations come from the validation framework. The *@NotNull* annotations make sure that the name and phone are not null. And *@Valid* annotation makes sure that *Name* and *Phone* objects are also validated.

It can be observed that we also added a default constructor to the *User* class with package visibility to satisfy the JPA requirements. However, it still sticks to the domain model. Remember that the domain model is technology-agnostic. It should remain the same with or without Spring Boot.

The following is the decorated *Name* class:

```
@Embeddable
public class Name implements Serializable {
      private static final long serialVersionUID =
-5928119910862624448L;
      @Column(name = "name")
      @Size(min = 8, max = 32)
```

```
    @NotBlank
    private String value;
    Name() {}
    ...
}
```

As you can recollect, it follows the immutable *ValueObject* pattern. The annotation *@Embeddable* marks the properties of *Name* to be part of another table, which is *UMS_USERS* in our case. To satisfy the JPA requirements, the *Name* is made to implement the *Serializable* interface along with a package-specific default constructor. The remaining decorations are self-explanatory.

The following is the decorated *PhoneNumber* class and there is nothing special about it. This is also an immutable *ValueObject* with appropriate JPA and validation annotations.

```
@Embeddable
public class PhoneNumber {
    @Column(name = "phone")
    @Min(9000000000L)
    private long value;
    PhoneNumber() {}
    ...
}
```

Finally, we come to the *UserRepository*. SpringBoot generates repositories with standard capabilities to save, find and delete the objects based on mere interface definition. You need not really implement this interface. Here is the decorated *UserRepository*:

```
@Repository
public interface UserRepository extends JpaRepository<User,
Name> {
}
```

We are done with implementing the domain layer of *AddService*.

Application layer

As you know, the application layer acts as the intermediary between the domain layers and the external clients, like REST clients in our case. It is responsible for receiving data transfer objects (DTOs) from the client and delegating the work to the appropriate domain objects. It is also responsible for doing the opposite while rendering the response.

The application layer of *AddService* consists of *UserController* along with a couple of DTOs. The DTOs do not require any sort of decoration.

Here is the decorated *UserController* with implementation:

```
@RestController
public class UserController {
    @Autowired
    private UserRepository repo;
    @Autowired
    private Validator validator;
    @PostMapping("/user")
    public ResponseEntity<UserRecord> add(@RequestBody
NewUser newUser) {
        Name name = new Name(newUser.name);
        PhoneNumber phoneNumber = new PhoneNumber(newUser.
phone);
        User user = new User(name, phoneNumber);
        Set<ConstraintViolation<User>> violations = validator.
validate(user);
        ResponseEntity<UserRecord> response;
        if (!violations.isEmpty())
            response = new ResponseEntity<>(HttpStatus.BAD_
REQUEST);
        else {
            repo.save(user);
            UserRecord record = new UserRecord(
                user.getName().getValue(),
                user.getPhone().getValue(),
                user.getSince());
            response = new ResponseEntity<>(record,
HttpStatus.OK);
        }
        return response;
    }
}
```

The annotation *@RestController* makes the *UserController* capable of handling the HTTP and JSON out-of-the-box. This is injected with *UserRepository* and Validator for the obvious purpose.

The annotation *@PostMapping("/user")* maps the POST REST API to the *add()* method. The Spring Boot takes care of marshalling the HTTP Request POST body to the *NewUser* DTO because of the *@RequestBody* annotation. The implementation of the *add()* method is self-explanatory.

Note that we skipped the messaging feature in the controller. We will refactor the controller to integrate it with a message broker like Kafka in the future.

Bootstrapping

And, finally, the *AddService* acts as the bootstrap for the whole microservice. As already discussed, the Spring Boot microservices are modelled as self-sufficient Java applications with a *main()* method that initialises everything, including an embedded Web server. The *@SpringBootApplication* annotations are just enough to get all the required bootstrap.

```
@SpringBootApplication
public class UserAddService {
      public static void main(String[] args) {
        SpringApplication.run(UserAddService.class, args);
      }
}
```

Configuration

Apart from the code we have seen so far, we also need to supply configuration. Though Spring Boot prefers convention over configuration, it works only for defaults. Whenever we want to override the defaults, we need to supply those overrides as configurations in the *application.properties* file. Here is the configuration for *AddService*:

```
spring.datasource.url=jdbc:h2:mem:glarimy
spring.jpa.database-platform=org.hibernate.dialect.H2Dialect
spring.h2.console.enabled=true
```

This configuration enables a Web based GUI console to interact with the H2 database. It also indicates the hibernate DDL processor to create a database named *glarimy* as per the annotations.

That's all.

A well-designed Spring Boot application thus requires these basic constructs: domain objects, repositories, DTOs, and controllers. This is exactly what domain-driven design proposed, isn't it? With the powerful annotations of Spring Boot, we get all the technological support like REST, persistence, validations, etc, without deviating from this design.

Building and running the microservice

Let us build the microservice and package it with the following command:

```
mvn clean package
```

It generates a fat JAR named *ums-add-service-0.0.1-SNAPSHOT.jar* under the *target* folder. Run the application using the usual Java command. You don't require any special skills and tools to run the microservice.

```
java -jar target/ums-add-service-0.0.1-SNAPSHOT.jar
```

Observe the console, and wait till the service is up and running. Since we do not have any Web interface for this service, you can use CURL to post a new user to the system.

However, just before doing that, verify the database by opening the link to the H2 database console at the URL *http://localhost:8080/h2-console*. Enter *jdbc:h2:mem:glarimy* as the JDBC URL in the dialogue and click *Connect.* Find the database table named *UMS_USERS* and click on it. It displays the query *SELECT * from UMS_USERS*. You do not get any records by running the query as we are yet to post any users.

Now open a terminal and run the *CURL* command to add a new user:

```
curl -X POST -H 'Content-Type: application/json' -i http://
localhost:8080/user --data '{"name":"Krishna Mohan",
"phone":9731423166}'
```

You should receive a *success* response. You can also check the database console to find the newly added record, as shown in Figure 5.

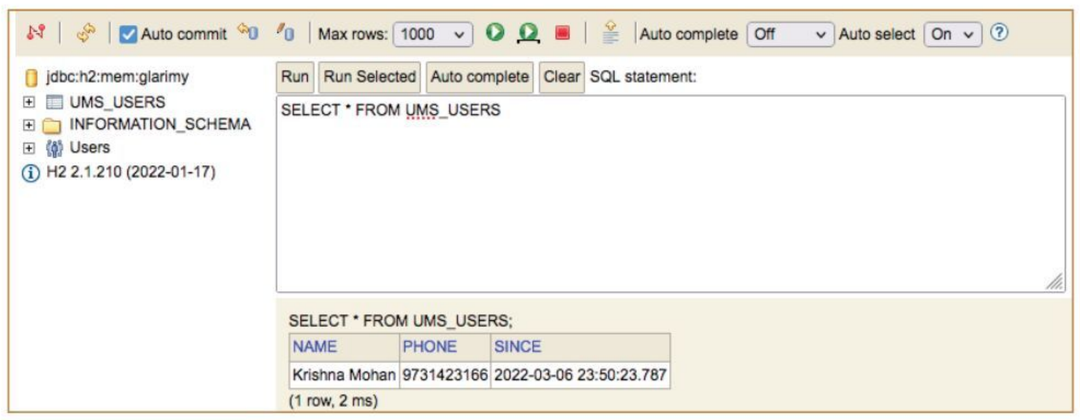


Figure 5: Results in the H2 database console

Road ahead

This part of 'The Design Odyssey' series illustrates a way to build microservices on the Java platform using the Spring Boot framework. It demonstrates that microservices are essentially modelled based on the concepts we have covered under domain-driven design. And the fundamental object-oriented patterns are sprinkled all over, thanks to the plumbing done by the framework. The biggest difference between a traditional Java Web application and a modern Java microservice is that we no longer package them as WAR files; instead, we package them as standard JAR files with an embedded Web server.

However, the Spring Boot microservice cannot be termed as self-sufficient yet, as it still requires a JRE pre-installed. Installing a JRE on every machine where the microservice is to be deployed is quite a task. How about including the JRE also into the deployment unit so that it truly becomes self-sufficient?

That's exactly what containerisation is. In the next part of this series of articles, we will containerise the Spring Boot service using Dockers and deploy it on a Kubernetes cluster. END

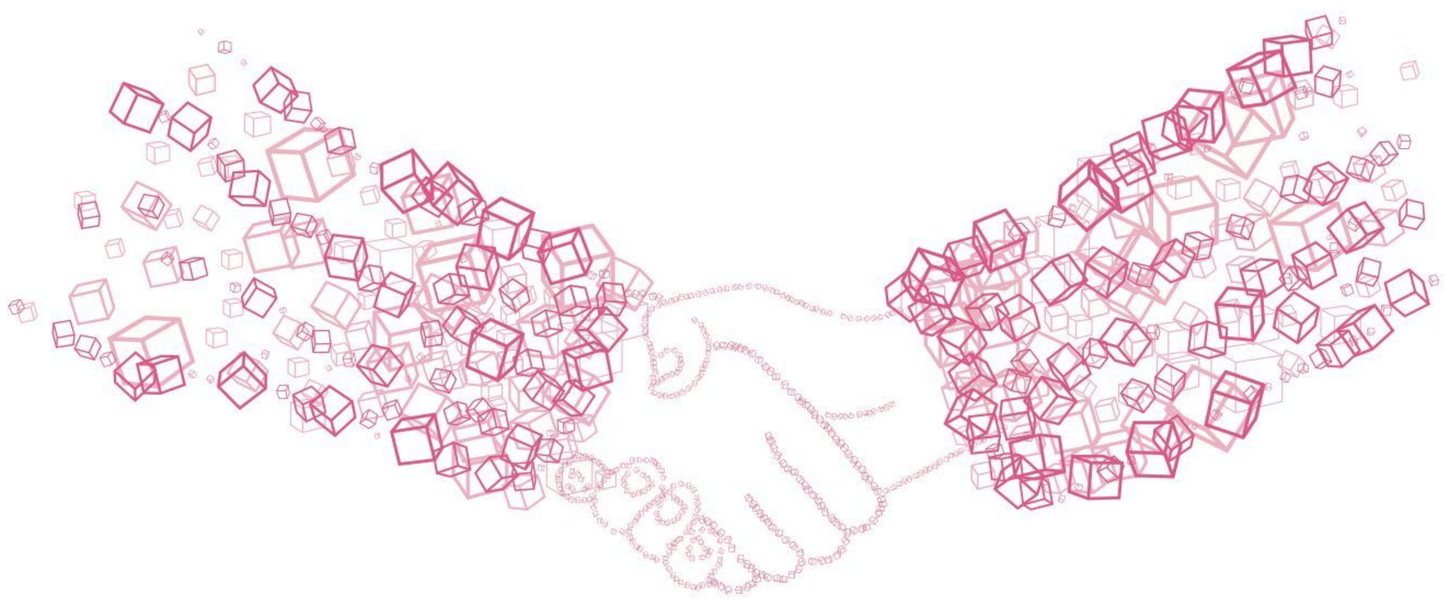
By: Krishna Mohan Koyya

The author is the founder and the principal technology consultant at Glarimy Technology Services, Bengaluru. He has delivered more than 500 programs to corporate clients on design and architecture in the last 12 years. Prior to that, he held various positions in the industry and academia for a decade.

FOCUS

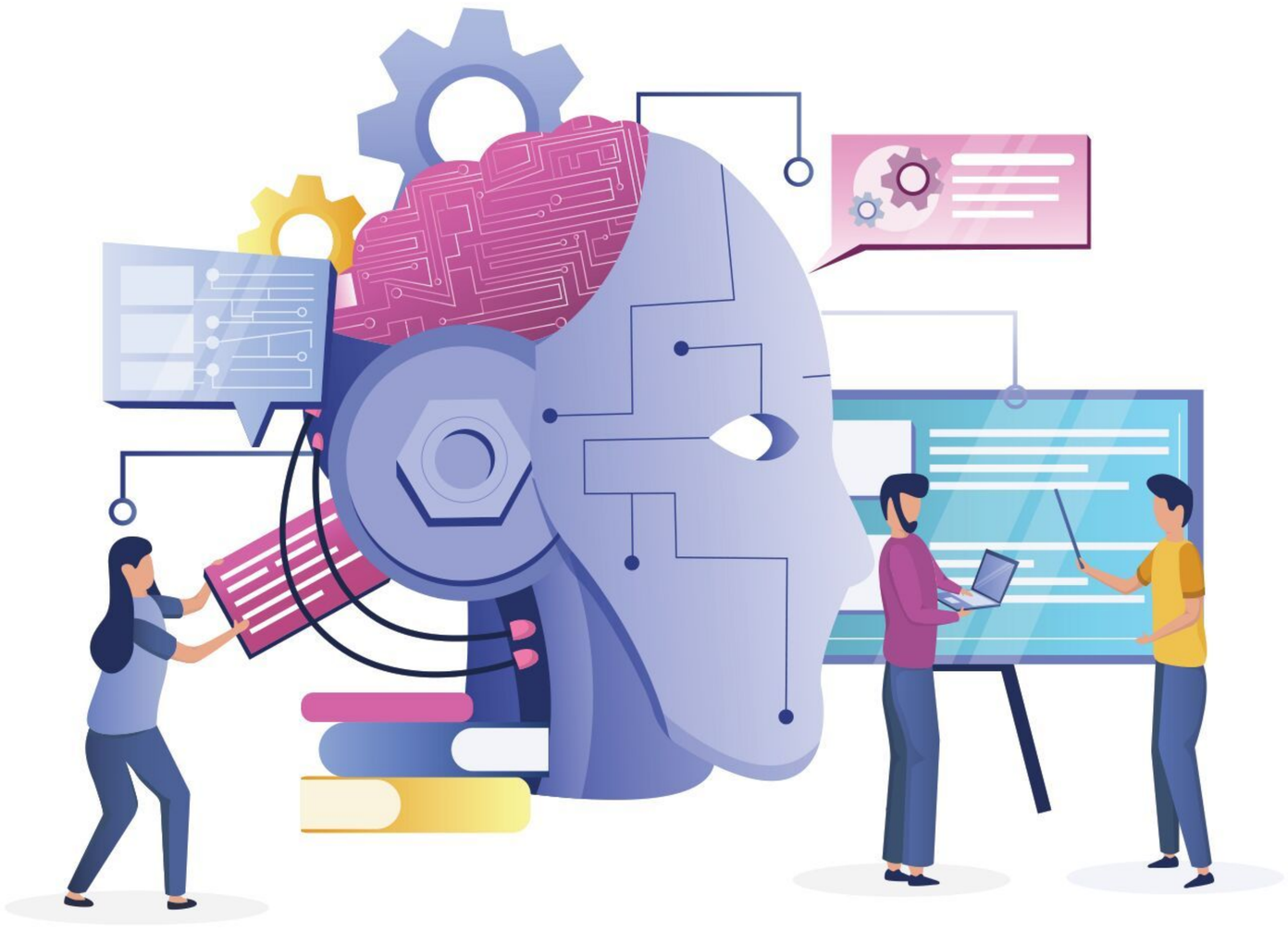
Programming With Scripts

DAML: The Programming Language for Smart Contracts in a Blockchain



The DAML smart contract language is a purpose-built domain specific language designed to encode the shared business logic of an application. It is used for the development and deployment of distributed applicationsin the blockchain environment. *Read more on page.....44*

Machine Learning: Classification Using Python



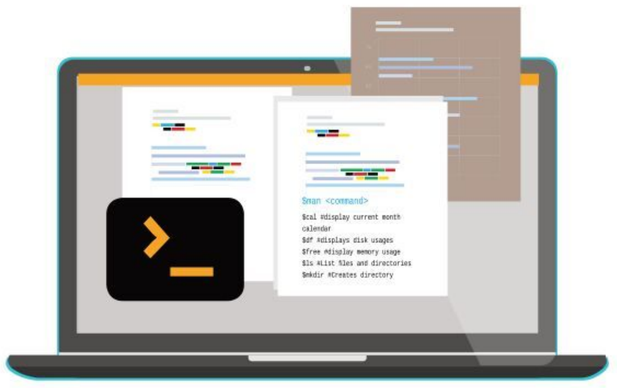
In machine learning (ML), a set of data is analysed to predict a result. Python is considered one of the best programming language choices for ML. In this article, we will discuss machine learning with respect to classification using Python. *Read more on page.....51*

Python 3: What's New?

Python is a programming language that allows us to communicate with computers. It is a means to tell a computer what it should do. It's a language that the computer understands and knows how to process. Let's take a quick look atwhat's new in the Python 3 release.

Read more on page.....47

The Basic Concepts of Shell Scripting



If you want to automate regular tasks and make your life easier, using shellscripts is a good option. This article introduces you to the basic concepts that will help you to write efficient shell scripts.

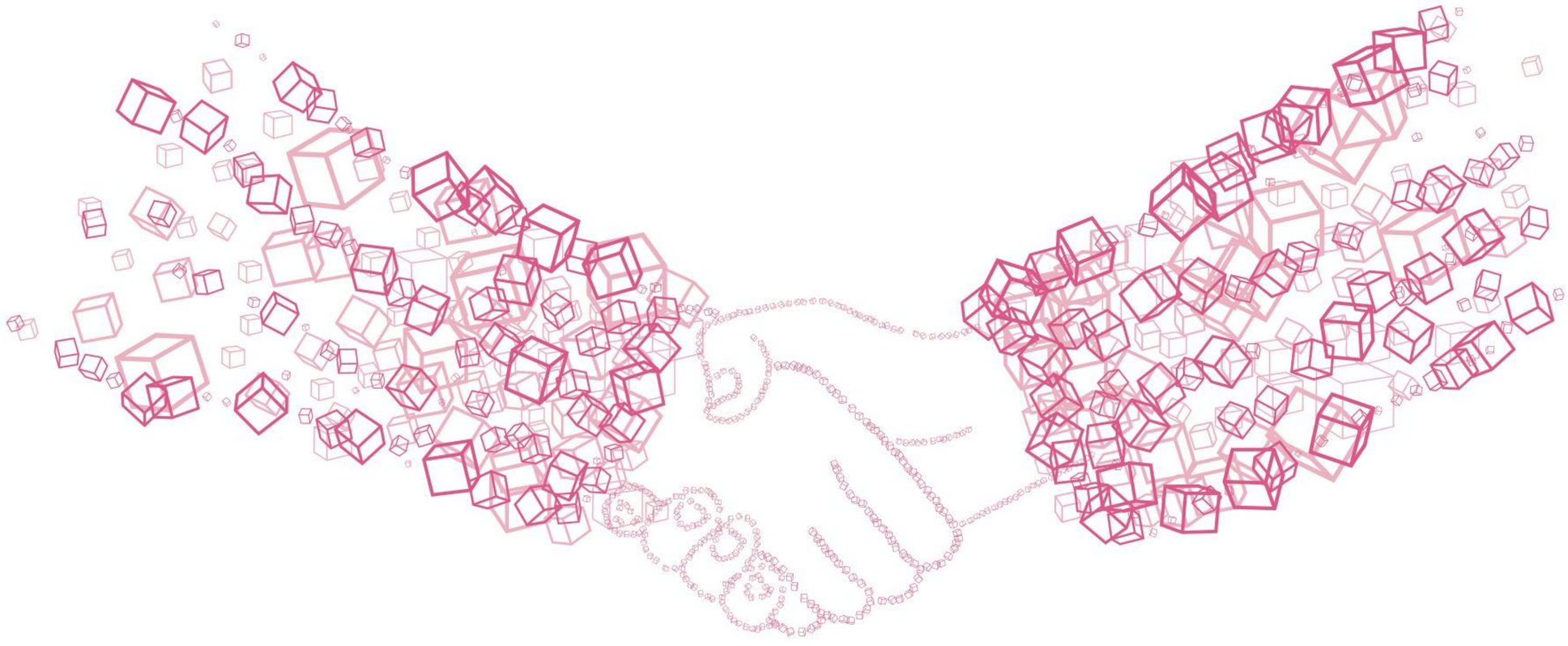
Read more on page.....48

Plotting Data in R: Graphs

R has a number of packages for plotting graphs and data visualisation, such as graphics, lattice, and ggplot2. In this ninth article in the R series, we shall explore the various functions to plot data in R.

Read more on page.....57

DAML: The Programming Language for Smart Contracts in a Blockchain



The DAML smart contract language is a purpose-built domain specific language designed to encode the shared business logic of an application. It is used for the development and deployment of distributed applications in the blockchain environment.

Blockchain technology is a secure mechanism to keep track of information in a way that makes it hard or impossible to modify or hack it. A blockchain integrates the digital ledger of transactions, which is copied and sent to every computer on its network. In each block of the chain, there are a number of transactions. When a new transaction takes place on the blockchain, a record of that transaction is added to the ledgers of everyone who is part of the chain.

Blockchain uses distributed ledger technology (DLT), in which a database isn't kept in one server or node. In a blockchain, transactions are recorded with an immutable cryptographic sign known as a hash. This means that if one block in one channel or chain is changed, it will be hard for hackers to change that block in the chain, as they would have to do this for every single version of the chain that is out there. Blockchains, like Bitcoin and Ethereum, keep growing as new blocks are added to the chain, which makes the ledger safer.

With the implementation of smart contracts in blockchain, there is automatic execution of scenarios without any human intervention. Smart contract technology makes it possible to enforce the highest level of security, privacy and anti-hacking implementations.

The use cases and applications of blockchain are:

- Cryptocurrencies
- Smart contracts
- Secured personal information
- Digital health records
- E-governance
- Non-fungible tokens (NFTs)
- Gaming
- Cross-border financial transactions
- Digital voting
- Supply chain management

As per *Statista.com*, the size of the blockchain technology market is increasing at a very fast speed since the last few years and is predicted to touch US$ 40 billion by 2025.

Programming languages and toolkits for blockchain

A number of programming languages and development toolkits are available for distributed applications and smart contracts. Programming and scripting languages for the blockchain include Solidity, Java, Vyper, Serpent, Python, JavaScript, GoLang, PHP, C++, Ruby, Rust, Erlang, etc, and are employed depending upon the implementation scenarios and use cases.

The choice of a suitable platform for the development and deployment of a blockchain depends on a range of factors including the need for security, privacy, speed of transactions and scalability (Figure 2).

The main platforms for the development of blockchain are:

- Ethereum
- XDC Network
- Tezos
- Stellar
- Hyperledger
- Ripple
- Hedera Hashgraph
- Quorum
- Corda
- NEO
- OpenChain
- EOS
- Dragonchain
- Monero

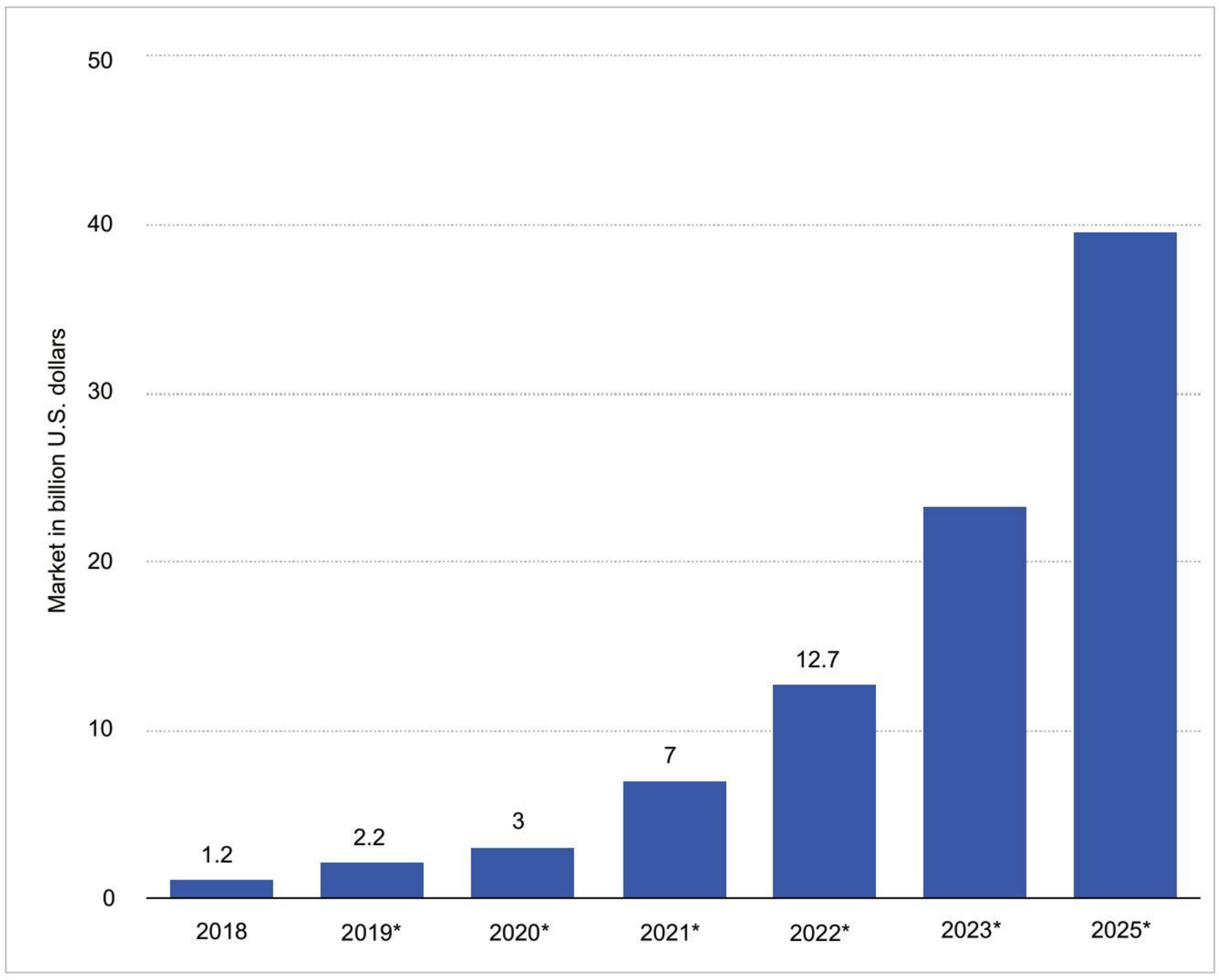


Figure 1: Market size of blockchain technology (Source: *Statista.com*)

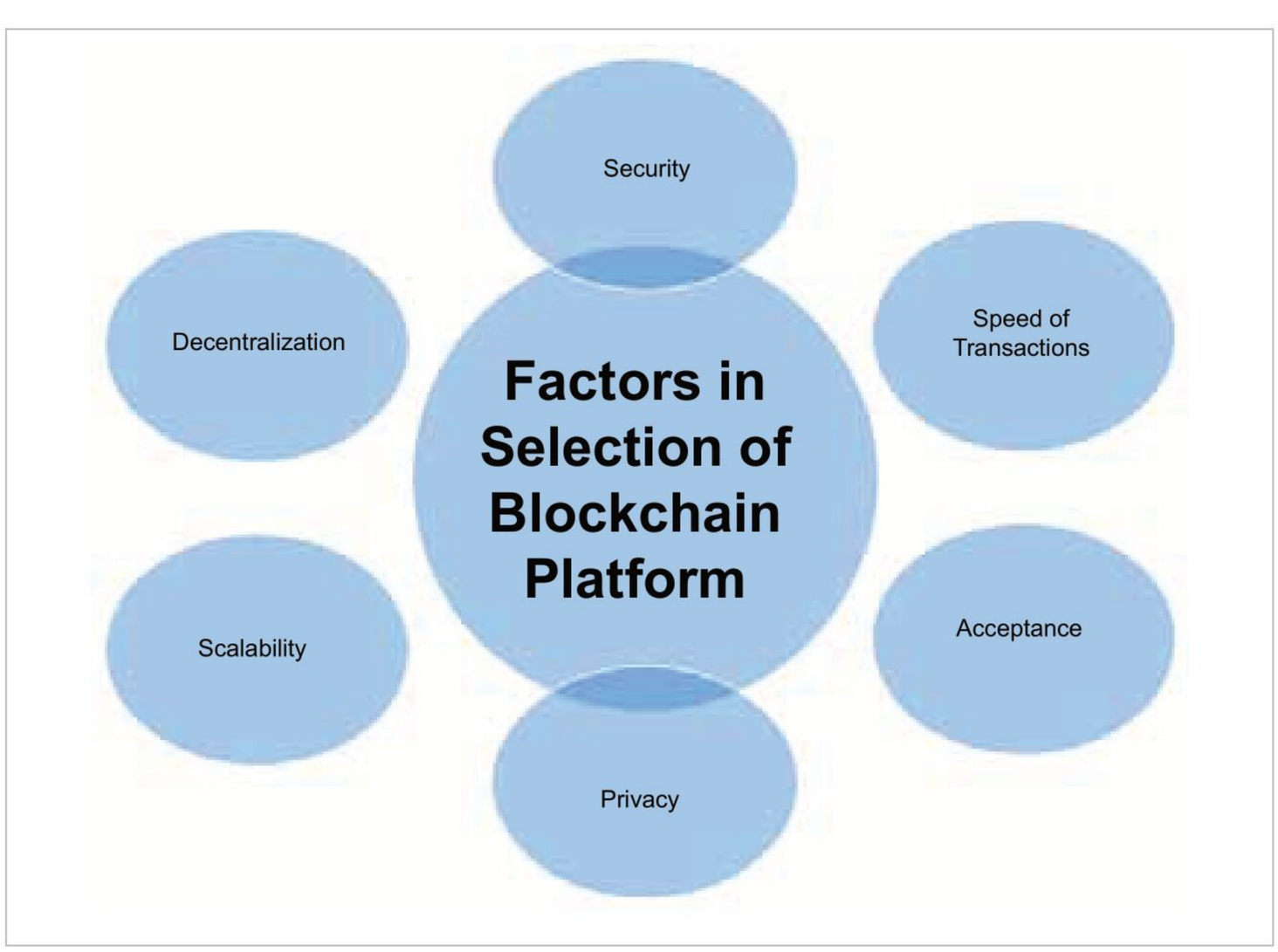


Figure 2: Factors to look at when selecting a blockchain platform

DAML: A high performance programming language

Digital Asset Modeling Language or DAML *(daml.com)* is a high performance programming language for the development and deployment of distributed applications in the blockchain environment. It is a lightweight and concise platform for rapid applications development.

The key features of DAML are:

- Fine-grained permissions
- Scenario based testing
- Data model
- Business logic
- Deterministic execution
- Storage abstraction
- No double spends
- Accountability tracking
- Atomic composability
- Authorisation checks
- Need-to-know privacy

Installation and working with DAML

The DAML SDK can be installed on Linux, macOS or Windows. The detailed instructions for installing DAML on multiple operating systems are available at *https://docs.daml.com/getting-started/installation.html*.

You must have the following to work with DAML:

- Visual Studio Code
- Java Development Kit (JDK)

DAML can be installed on Windows by downloading and running the executable installer available at *https://github.com/digital-asset/daml/*

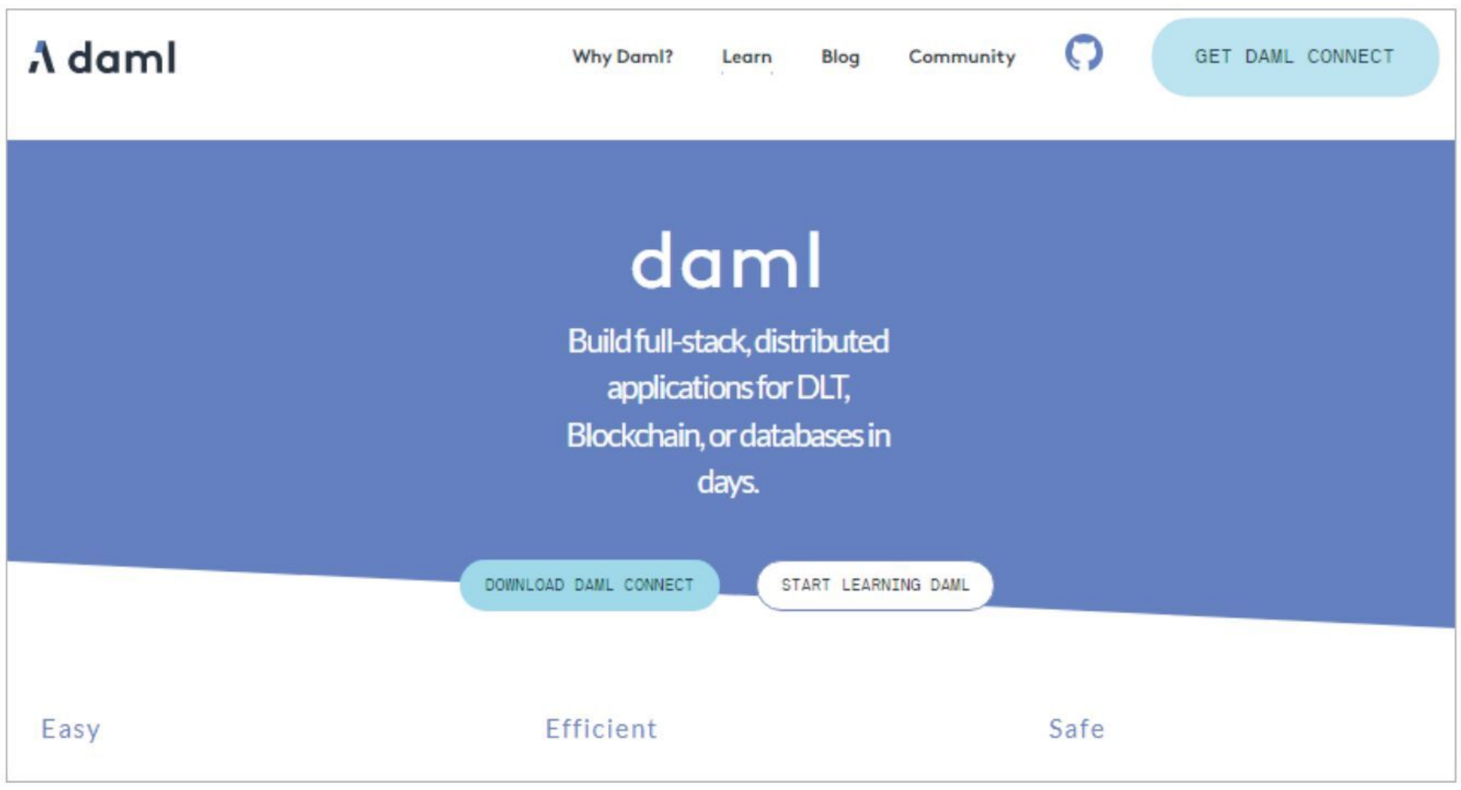


Figure 3: Official portal of DAML

```
C:\Users\Gaurav Kumar>daml new myapp --template create-daml-app
Created a new project in "myapp" based on the template "create-daml-app".


C:\Users\Gaurav Kumar>daml new --list
```

Figure 4: Creating a new app

```
C:\Users\Gaurav Kumar>cd myapp

C:\Users\Gaurav Kumar\myapp>daml start

2022-02-26 04:22:54.91 [INFO]  [build]
Compiling myapp to a DAR.

2022-02-26 04:22:56.78 [INFO]  [build]
Created .daml\dist\myapp-0.1.0.dar
Generating 057eed1fd48c238491b8ea06b9b5bf85a5d4c9275dd3f6183e0e6b01730cc2
ba
Generating 108aa371793a39379d09dd6f02b5613e8bf37ec9893a826747545df5318475
f1
Generating 38e6274601b21d7202bb995bc5ec147decda5a01b68d57dda422425038772a
f7
Generating 3f4deaf145a15cdcfa762c058005e2edb9baa75bb7f95a4f8f6f937378e864
15
```

Figure 5: Running DAML



Figure 6: Login panel in DAML app

releases/download/v1.18.1/daml-sdk-1.18.1-windows.exe.

Installation of DAML on Linux or Mac can be done by executing the following in the terminal:

```
$ curl -sSL https://get.daml.com/ | sh
```

After installation of DAML, the new blockchain based app can be created, as shown in Figures 4 and 5.

In another terminal, the new app is navigated and project dependencies are installed:

```
WorkingDirectory>cd myapp/ui
WorkingDirectory>npm install
WorkingDirectory>npm start
```

The WebUI is started and the app is accessed on the Web browser with the URL *http://localhost:3000/*.

Scope for research and development

Blockchain technology has a wide range of development platforms and frameworks for different categories of applications. Many of these platforms are free and open source, which can be downloaded and deployed for research based implementations. Research scholars, practitioners and academicians can use these platforms to propose and implement their algorithms for numerous applications. END

References

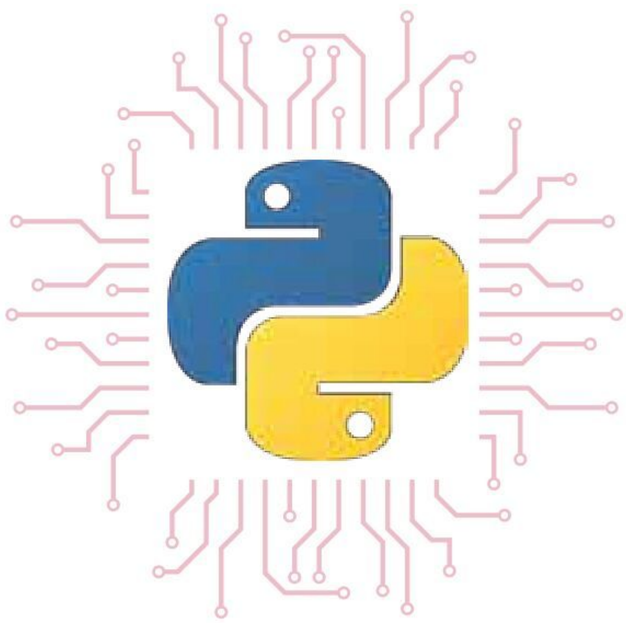
[1] DAML official portal, *https://daml.com/*
[2] Statista research portal, *https://www.statista.com/statistics/647231/worldwide-blockchain-technology-market-size/*

By: Dr Gaurav Kumar

The author is associated with various academic and research institutes, where he delivers expert lectures and conducts technical workshops on the latest technologies and tools.

Python 3: What's New?

Python is a programming language that allows us to communicate with computers. It is a means to tell a computer what it should do. It's a language that the computer understands and knows how to process. Let's take a quick look at what's new in the Python 3 release.





Learning a programming language means understanding how the language works, its rules, grammar, syntax and semantics. This is similar to the way we learn any language we speak.

Why should one learn Python?

Python is the most popular open source programming language currently. The language is relatively quick to learn, and simple to read and understand. It has intuitive and powerful syntax with a vast array of libraries, and allows scripts to be executed in virtual environments. It is ubiquitous and used for machine learning, Web development, data science, game development, desktop applications, 3D modelling and scientific fields.

Technically, Python is:

- Object-oriented
- Dynamically typed
- Strongly typed
- White-space delimited
- Interpreted

What's new in Python 3

Python 3 has new keywords, which provide new features. The syntax has been improved, leading to better code patterns. This improves readability and makes expressions easier to use.

New libraries, packages, classes and functions have been introduced. Programmers can directly start using the available modules and libraries, leading to faster development and improved productivity.

It has enhanced support for automated testing, statistical modelling, concurrent processing, asynchronous features, network programming, debugging, etc.

There are improvements with respect to reliability, which eliminate hidden race conditions, enhance security, remove deterministic but surprising behaviours, and generally make creating robust Python applications much easier.

Key differences between Python 2 and Python 3

Python 3 has intentionally broken compatibility with Python 2. Table 1 lists a few differences between the two versions.

There are tools available to convert Python code from v2 to v3. '2to3' is part of the standard Python library (comes along with the Anaconda package) and is an effective tool for conversion. END

Table 1: The differences between Python 2 and Python 3

Function	Python 2		Python 3	
	Input	Output	Input	Output
Print	print "hi"	hi	print("hi")	hi
Division	print 7/2	3	print(7/2)	3.5
Bytes	print b'hello'	hello	print (b'hello')	b'hello'
Unicode	print type(u"hello")	<type 'unicode'>	print type(u"hello")	<class 'str'
Round	round(100.5)	101	round(100.5)	100
Input	raw_input("Enter name")		Input("Enter name")	

Reference

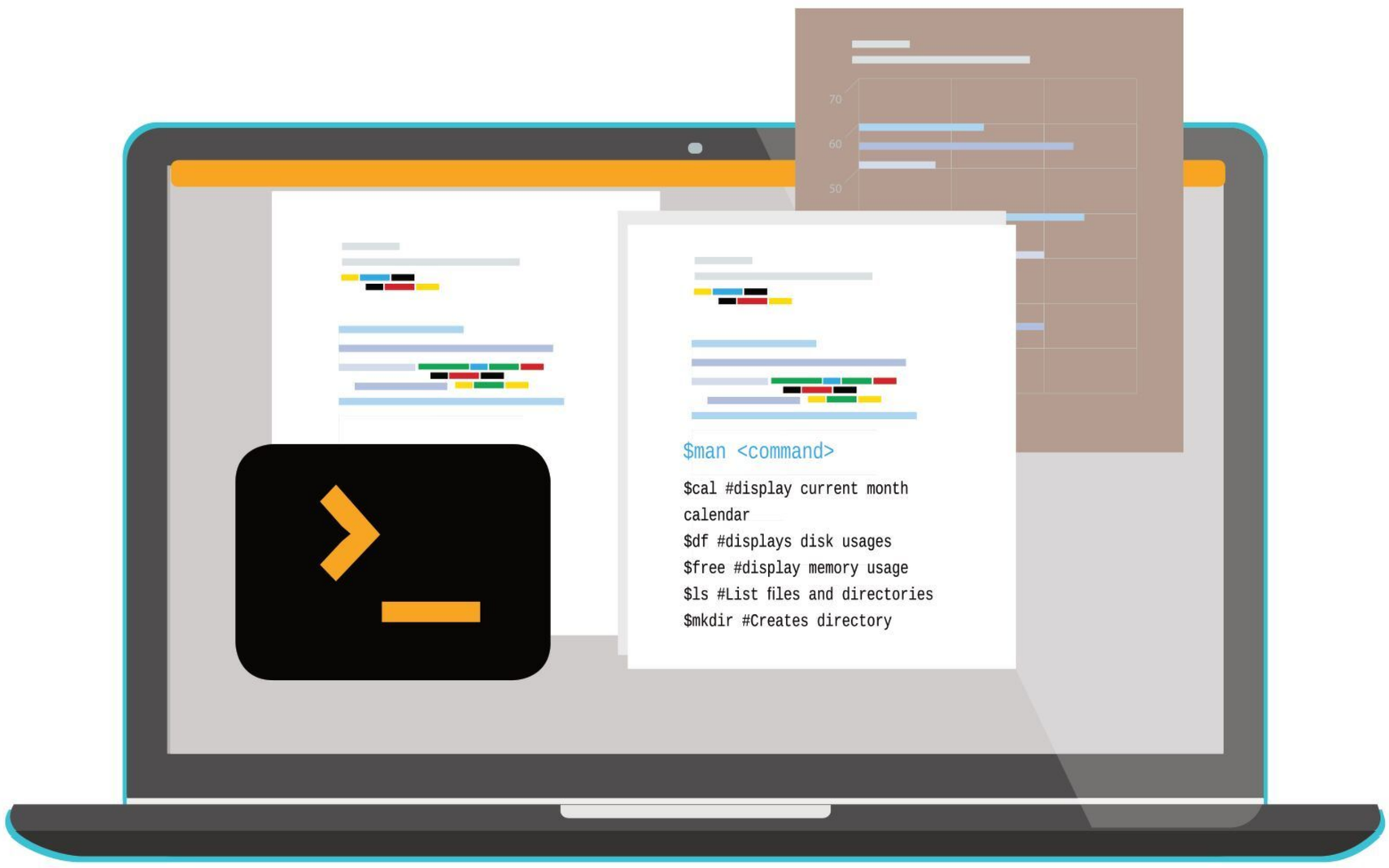
Docs.python.org/

By: Phani Kiran

The author has over 17 years of experience in the IT industry and has worked on Big Data technologies.

The Basic Concepts of Shell Scripting

If you want to automate regular tasks and make your life easier, using shell scripts is a good option. This article introduces you to the basic concepts that will help you to write efficient shell scripts.



A shell script is a computer program designed to be run by the UNIX shell, a command-line interpreter. The various dialects of shell scripts are considered to be scripting languages. Typical operations performed by shell scripts include file manipulation, program execution, and printing of text. A script that sets up the environment, runs the program, and does any necessary cleanup or logging, is called a wrapper.

Identification of shell prompt

You can identify whether the shell prompt on a Linux based computer is a normal or super user by looking at the symbols of the prompt in the terminal window. The '#' symbol is used for a super user and the '$' symbol is used for a user with standard privileges.

Basic commands

The script comes with too many commands that can be executed on the terminal window to manage your computer. Details of each command can be found in the manual included with the command. To view the manual, you need to run the command:

```
$man <command>
```

A few frequently used commands are:

```
$date #display current date and time
$cal #display current month calendar
$df #displays disk usages
$free #display memory usage
$ls #List files and directories
$mkdir #Creates directory
```

Each command comes with several options that can be used along with it. You can refer to the manual for more details. See Figure 1 for the output of:

```
$man date
```

```
DATE(1)                                User Commands                                DATE(1)

NAME
       date - print or set the system date and time

SYNOPSIS
       date [OPTION]... [+FORMAT]
       date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]

DESCRIPTION
       Display the current time in the given FORMAT, or set the system date.

       Mandatory arguments to long options are mandatory for short options too.

       -d, --date=STRING
              display time described by STRING, not 'now'

       --debug
              annotate the parsed date, and warn about questionable usage to stderr

       -f, --file=DATEFILE
              like --date; once for each line of DATEFILE

       -I[FMT], --iso-8601[=FMT]
              output  date/time  in  ISO  8601  format.  FMT='date' for date only (the de-
              fault), 'hours', 'minutes', 'seconds', or 'ns' for date and time to the  in-
              dicated precision.  Example: 2006-08-14T02:34:56-06:00

       -R, --rfc-email
              output date and time in RFC 5322 format.  Example: Mon, 14 Aug 2006 02:34:56
              -0600
```

Figure 1: Manual of *date* command

Redirection operators

The redirection operator is really useful when you want to capture the output of a command in a file or redirect to a file.

`$ls -l /usr/bin >file`	default stdout to file
`$ls -l /usr/bin 2>file`	redirects stderr to file
`$ls -l /usr/bin > ls-output 2>&1`	redirects stderr & stdout to file
`$ls -l /usr/bin &> ls-output`	redirects stderr & stdout to file
`$ls -l /usr/bin 2> /dev/null`	/dev/null bitbucket

Brace expansion

Brace expansion is one of the powerful options UNIX has. It helps do a lot of operations with minimal commands in a single line instruction. For example:

```
$echo Front-{A,B,C}-Back
Front-A-Back, Front-B-Back, Front-C-Back

$echo {Z..A}
Z Y X W V U T S R Q P O N M L K J I H G F E D C B A

$mkdir {2009..2011}-0{1..9} {2009..2011}-{10..12}
```

This creates a directory for 12 months from 2009 to 2011.

Environment variables

An environment variable is a dynamic-named value that can affect the way running processes will behave on a computer. This variable is a part of the environment in which a process runs.

printenv	Print part of all of the environment
set	set shell options
export	export environment to subsequently executed programs
alias	create an alias for command

Network commands

Network commands are very useful for troubleshooting issues on the network and to check the particular port connecting to the client.

ping	Send ICMP packets
traceroute	Print route packets to a network
netstat	print network connection, routing table, interface stats
ftp/lftp	Internet file transfer program
wget	Non Interactive network downloader
ssh	OpenSSH SSH Client (remote login program)
scp	secure copy
sftp	Secure File transfer program

Grep commands

Grep commands are useful to find the errors and debug the logs in the system. It is one of the powerful tools that shell has.

grep -h '.zip' file.list	. is any character
grep -h '^zip' file.list	starts with zip
grep -h 'zip$' file.list	ends with zip
grep -h '^zip$' file.list	containing only zip
grep -h '[^bz]zip' file.list	not containing b and z
grep -h '^[A-Za-z0-9]' file.list	file containing any valid names

Quantifiers

Here are some examples of quantifiers:

?	match element zero or one time
*	match an element zero or more times
+	Match an element one or more times
{}	match an element specfic number of times

Text processing

Text processing is another important task in the current IT world. Programmers and administrators can use the commands to dice, cut and process texts.

cat -A $FILE	To find any CTRL character introduced
sort file1.txt file2.txt file3.txt > final_sorted_list.txt	sort all files once
ls - l | sort -nr -k 5	key field 5th column
sort --key=1,1 --key=2n distor.txt	key field 1,1 sort and second column sort by numeric
sort foo.txt | uniq -c	to find repetition
cut -f 3 distro.txt	cut column 3
cut -c 7-10	cut character 7 - 10
cut -d ':' -f 1 /etc/password	delimiter :
sort -k 3.7nbr -k 3.1nbr -k 3.4nbr distro.txt	3 rd field 7 the character, 3rd field 1 character
paste file1.txt file2.txt > newfile.txt	merge two files
join file1.txt file2.txt	join on common two fields

Hacks and tips

In Linux, we can go back to our history of commands by either using simple commands or control options.

clear	clears the screen
history	stores the history
script filename	capture all command execution in a file

Tips:

```
History    : CTRL + {R, P }
!!number : command history number
!!              : last command
!?string    : history containing last string
!string      : history containing last string
export HISTCONTROL=ignoredups
export HISTSIZE=10000
```

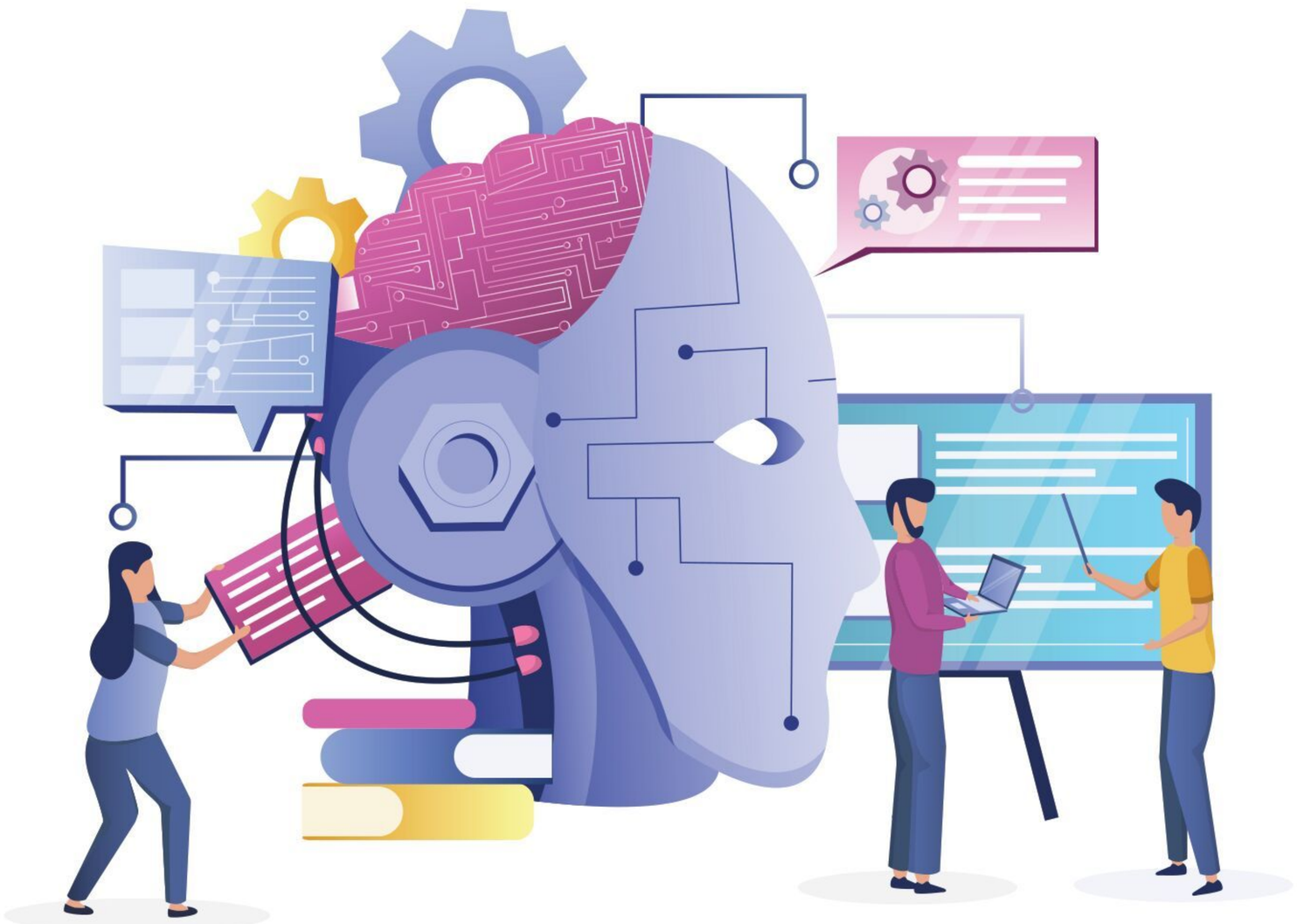
As you get familiar with the Linux commands, you will be able to write wrapper scripts. All manual tasks like taking regular backups, cleaning up files, monitoring the system usage, etc, can be automated using scripts. This article will help you to start scripting, before you move to learning advanced concepts. END

By: Sathya Narayanan

The author is a computer science engineer. In the past 15+ years, he has worked in many IT organisations including TechMahindra, Cognizant Technology Solutions, and HCL-Cisco Technologies. He is currently working at Akamai Technologies as a software engineer.

Machine Learning: Classification Using Python

In machine learning (ML), a set of data is analysed to predict a result. Python is considered one of the best programming language choices for ML. In this article, we will discuss machine learning with respect to classification using Python.



Let's say you want to teach a child to differentiate between apples and oranges. There are various ways to do this. You could ask the child to touch both kinds of fruits so that they get familiar with the shape and the softness. You could also show her multiple examples of apples and oranges, so that they can visually spot the differences. The technological equivalent of this process is known as machine learning.

Machine learning teaches computers to solve a particular problem, and to get better at it through experience. The example discussed here is a classification problem, where the machine is given various labelled examples, and is expected to label an unlabelled sample using the knowledge it acquired from the labelled samples. A machine learning problem can also take the form of regression, where it is expected to predict a real-valued solution to a given problem based on known samples and their solutions. Classification and regression are broadly termed as supervised learning. Machine learning can also be unsupervised, where the machine identifies patterns in unlabelled data, and forms clusters of samples with similar patterns. Another form of machine learning is reinforcement learning, where the machine learns from its environment by making mistakes.

Classification

Classification is the process of predicting the label of a given set of points based on the information obtained from known points. The class, or label, associated with a data set could be binary or multiple in nature. As an example, if we have

Table 1

Age	Credit rating	Job	Property owned	Loan approved
35	good	yes	yes	**yes**
32	poor	yes	no	**no**
22	fair	no	no	**no**
42	good	yes	no	**yes**

to label the sentiment associated with a sentence, we could label it as positive, negative or neutral. On the other hand, problems where we have to predict whether a fruit is an apple or an orange will have binary labels. Table 1 gives a sample data set for a classification problem.

In this table, the value of the last column, i.e., loan approved, is expected to be predicted based on the other variables. In the subsequent sections, we will learn how to train and evaluate a classifier using Python.

Training and evaluating a classifier

In order to train a classifier, we need to have a data set containing labelled examples. Though the process of cleaning the data is not covered in this section, it is recommended that you read about various data preprocessing and cleaning techniques before feeding your data set to a classifier. In order to process the data set in Python, we will import the pandas package and the data frame structure. You may then choose from a variety of classification algorithms such as decision tree, support vector classifier, random forest, XG boost, ADA boost, etc. We will look at the random forest classifier, which is an ensemble classifier formed using multiple decision trees.

```
from sklearn.ensemble import
RandomForestClassifier
from sklearn import metrics

classifier = RandomForestClassifier()

#creating a train-test split with a
proportion of 70:30
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.33)

classifier.fit(X_train, y_train) #train
the classifier on the training set

y_pred = classifier.predict(X_test)
#evaluate the classifier on unknown data

print("Accuracy: ", metrics.accuracy_
score(y_test, y_pred)) #compare the
predictions with the actual values in
the test set
```

Although this program uses accuracy as the performance metric, a combination of metrics should be used, as accuracy tends to generate non-representative results when the test set is imbalanced. For instance, we will get a high accuracy if the model gives the same prediction for every record and the data set that is used to test the model is imbalanced, i.e., most of the records in the data set have the same class that the model predicted.

Tuning a classifier

Tuning refers to the process of modifying the values of the hyperparameters of a model in order to improve its performance. A hyperparameter is a parameter whose value can be changed to improve the learning process of the algorithm.

The following code depicts random search hyperparameter tuning. In this, we define a search space from which the algorithm will pick different values, and choose the one that produces the best results:

```
from sklearn.model_selection import
RandomizedSearchCV

#define the search space
min_samples_split = [2, 5, 10]
min_samples_leaf = [1, 2, 4]
grid = {'min_samples_split' : min_
samples_split, 'min_samples_leaf' :
min_samples_leaf}

classifier =
RandomizedSearchCV(classifier, grid,
n_iter = 100)

#n_iter represents the number of
samples to extract from the search
space
#result.best_score and result.best_
params_ can be used to obtain the best
performance of the model, and the best
values of the parameters

classifier.fit(X_train, y_train)
```

Voting classifier

You can also use multiple classifiers and their predictions to create a model that will give a single prediction based on the individual predictions. This process (in which only the number of classifiers that voted for each prediction is considered) is called hard voting. Soft voting is a process in which each classifier generates a probability of a given record belonging to a particular class, and the voting classifier generates as its prediction, the class that obtained the maximum probability.

A code snippet for creating a soft voting classifier is given below:

```
soft_voting_clf = VotingClassifier(
        estimators=[('rf', rf_clf),
('ada', ada_clf), ('xgb', xgb_clf),
('et', et_clf), ('gb', gb_clf)],
        voting='soft')
    soft_voting_clf.fit(X_train, y_
train)
```

This article has summarised the use of classifiers, tuning a classifier and the process of combining the results of multiple classifiers. Do use this as a reference point and explore each area in detail. END

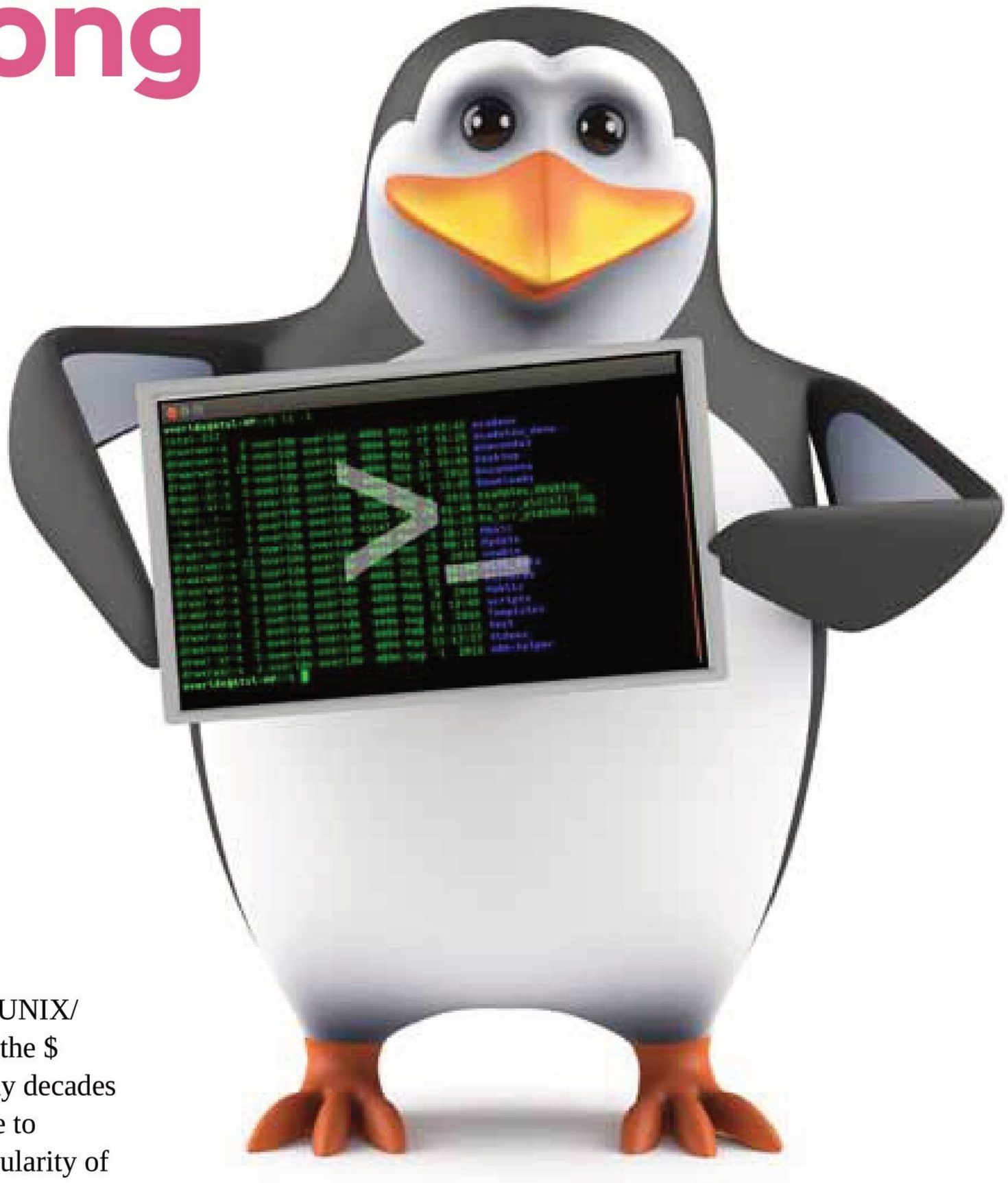
Reference

Jordan, M.I., & Mitchell, T.M. (2015), Machine learning: Trends, perspectives, and prospects. Science, 349(6245), 255-260.

By: Gayatri Venugopal

The author is a faculty member at a private university, and is currently pursuing her PhD in the areas of machine learning and natural language processing.

Shell Scripting is Still Going Strong



This article introduces you to the basics of shell scripting and its importance in day-to-day life. A shell script is a command-line interpreter that runs on a UNIX/ Linux shell.

The first thing we notice when we log into a UNIX/ Linux system is the blinking cursor next to the $ sign. This is the shell. It has been – for many decades – the ubiquitous (and many times the only) interface to interact with a computer. Before the advent and popularity of graphical user interfaces (GUIs), the terminal and the shell were the only mechanism to make the computer do what we wanted it to do. At first glance, one may wonder what the shell does – other than passing commands to the underlying operating system for execution. Most of us are familiar with commands like 'ls' (for listing contents of a directory), 'cd' (for changing the current directory), and so on. It is through the shell that we can execute these commands. The shell understands the text we type — converts it into tokens — and then executes them on the operating system.

Flavours

Initially, the terminal started with the humble Bourne shell or 'sh'. Over the years, many shell variants were developed and used. Some of the popular ones are 'C Shell' / 'csh' and 'Korn Shell' / 'ksh'. 'sh' fell out of favour for a few years, but has gained popularity once again through its recent avatar, namely 'bash' / 'Bourne Again Shell'.

What does the shell actually do?

The shell is the immediate interface between the operating system (OS) and the user. We make the computer do what we want, by using commands and applications supported by the tools installed on the computer we are using. Some commands are applications installed on the operating system, while some are built into the shell itself. Some of the commands built into bash are 'clear', 'cd', 'eval', and 'exec', to name a few, while commands like 'ls', and 'mkdir' are applications. The commands built into the shell vary as per the shell.

In this article, we cover a few aspects related to 'bash'.

More about the shell

Most of us have used commands like 'ls', 'cd', and 'mkdir'. When we run the 'ls -l' command on a directory, all the directories and files in that directory are listed on the screen.

If the number is large, the screen scrolls. If the terminal does not support scroll bars (as was the case for many years), there is no way to look at the entries that have scrolled past. To help overcome this, we use commands like 'more' and 'less'. These allow us to view the output on a page-by-page basis. The command typically used is:

```
ls -l | less
```

What is the shell doing here? What looks like a single command is actually two commands executing one after the other, *ls* and *less*. The pipe ('|') connects the two programs, but the connection is managed by the shell. Because of the pipe character, the shell connects the two programs – it connects the standard output of the *ls* command and connects it to the standard input or standard in or stdin of *less*. The pipe feature allows us to take the output of any program and provide it as the input to another program – without us having to do any changes to the programs. This is the philosophy of many UNIX/Linux applications — keep the applications simple and then combine many applications together to achieve the end result, rather than having one program do many things.

If needed, we can redirect the output of *ls* to a file and then view it using 'vi'. For this, we use the command:

```
ls -l  > /tmp/my_file.txt
vi /tmp/my_file.txt
```

In this case, the output of *ls* is being redirected to a file. This is managed by the shell, which understands the '>' symbol to mean redirection. It treats the token that follows as a file.

Automation using shell

This ability to combine commands is one of the key elements for the creation of automation scripts using shell commands. In my most recent project, we were executing Python/Spark (PySpark) applications using cluster mode. Each application executed many structured query language (SQL) statements – SparkSQL. To keep track of application progress, we were printing details about the SQL being executed. This allowed us to maintain a log of what was happening in the application. As the applications were executed in cluster mode, to view the log, we had to use the *yarn* command as follows:

```
yarn log -applicationId [application_id]
```

In most cases, the log produced by an application was very large. So we typically piped the log to 'less' or redirected it to a file. The command we used was:

```
yarn log -aplicationId [application_id] | less
```

Our development team had a strength of 40 people. Each one had to remember this command. To make it simpler, I converted this command into a bash script. For this, I created a file with a '.sh' extension. On UNIX and Linux systems, file extension does not matter. As long as the file is an executable, it will work. Extensions have significance on MS Windows.

Important thing to remember

The shell is an interpreter. This means that it will read the program line by line and execute it. The limitation of this approach is that errors (if any) are not identified upfront. Errors are not identified till they are read and executed by the interpreter. In short, we can have a shell program that will execute perfectly for the first 20 lines and then fail due to a syntax error on line 21. When the script fails at line 21, the shell does not unroll/undo the previous steps. When such a thing occurs, we have to correct the script and start execution from the first line. Thus, as an example, if we have deleted a few files before encountering an error, execution of the shell script will stop, but the files are gone forever.

The script I created was:

```
#!/bin/bash
yarn log -applicationId 123 | less
```

…where 123 was the application ID.

The first two characters of the first line are magic characters. They tell the script that this is an executable file and the line contains the name of the program to be used for execution. The remaining lines of the script are passed to the program mentioned. In this case, we are going to execute bash. Even after including the first line, we have to apply execute permissions to the file using:

```
chmod +x my_file.sh
```

After giving execute permissions to the file, we can execute it as:

```
./my_file.sh
```

If we do not give execute permissions to the file, we can execute the script as:

```
sh ./my_file.sh
```

Passing parameters

You will realise quickly that such a script is handy, but becomes useless immediately. Each time we execute the Python/Spark application, a new ID is generated. Hence, for each run, we have to edit the file and add the new application ID. This

definitely reduces the usability of the script. To be useful, we should be passing the application ID as a parameter:

```
#!/bin/bash
yarn –log -applicationId ${1} | less
```

We need to execute the script as:

```
./show_log.sh 123
```

The script will execute the *yarn* command, fetch the log for the application and allow us to view it.

What if we want to redirect the output to a file? Not a problem. Instead of sending the output to *less*, we can redirect it to a file:

```
#!/bin/bash
ls -l ${1} > ${2}
view ${2}
```

To run the script, we have to provide two parameters, and the command becomes:

```
./my_file.sh /tmp /tmp/listing.txt
```

When executed, $1 will bind to */tmp* and $2 will bind to */tmp/listing.txt.* For the shell, the parameters are named from one to nine. This does not mean we cannot pass more than nine parameters to a script. We can, but that is the topic of another article. You will note that I have mentioned the parameters as ${1} and ${2} instead of $1 and $2. It is a good practice to enclose the name of the parameter in curly brackets as it allows us to unambiguously combine the parameters as part of a longer variable. For example, we can ask the user to provide file name as a parameter and then use that to form a larger file name. As an example, we can take $1 as the parameter and create a new file name as *${1}_student_names.txt.*

Making the script robust

What if the user forgets to provide parameters? The shell allows us to check for such conditions. We modify the script as below:

```
#!/bin/bash
if [ -z “${2}” ]; then
    echo “file name not provided”
    exit 1
fi
if [ -z “${1}” ]; then
    echo “directory name not provided”
    exit 1
fi
DIR_NAME=${1}
FILE_NAME=${2}
ls -l ${DIR_NAME} > /tmp/${FILE_NAME}
view /tmp/${FILE_NAME}
```

In this program, we check if the proper parameters are passed. We exit the script if parameters are not passed. You will note that I am checking the parameters in reverse order. If we check for the presence of the first parameter before checking the presence of the second parameter, the script will pass to the next step if only one parameter is passed. While the presence of parameters can be checked in ascending order, I recently realised that it might be better to check from nine to one, as we can provide proper error messages. You will also note that the parameters have been assigned to variables. The parameters one to nine are positional parameters. Assigning positional parameters to named parameters makes it easy to debug the script in case of issues.

Automating backup

Another task that I automated was that of taking a backup. During development, in the initial days, we did not have a version control system in place. But we needed to have a mechanism to take regular backups. So the best method was to write a shell script that, when executed, copied all the code files into a separate directory, zipped them and then uploaded them to HDFS, using the date and time as the suffix. I know that this method is not as clean as having a version control system, as we store complete files and finding differences still needs the use of a program like *diff*; however, it is better than nothing. While we did not end up deleting the code files, the team did end up deleting the *bin* directory where the helper scripts were stored!!! And for this directory I did not have a backup. I had no choice but to re-create all the scripts.

Once the source code control system was in place, I easily extended the backup script to upload the files to the version control system in addition to the previous method uploading to HDFS.

Summing up

These days, programming languages like Python, Spark, Scala, and Java are in vogue as they are used to develop applications related to artificial intelligence and machine learning. While these languages are far more powerful when compared to shells, the ‘humble’ shell provides a ready platform that allows us to create helper scripts that ease our day-to-day tasks. The shell is quite powerful, more so because we can combine the powers of all the applications installed on the OS. As I found out in my project, even after many

decades, shell scripting is still going strong. I hope I have convinced you to give it a try.

One for the road

Shell scripts can be very handy. Consider the following command:

```
spark3-submit --queue pyspark --conf “spark.yarn.principal=
abcd@abcd.com --conf “spark.yarn.keytab=/keytabs/abcd.keytab
--jars /opt/custom_jars/abcd_1.jar --deploy-mode cluster
--master yarn $*
```

We were expected to use this command while executing a Python/Spark application. Now imagine this command has to be used multiple times a day, by a team of 40 people. Most of us will copy this command in Notepad++, and each time we need to use it, we will copy it from Notepad++ and paste it on the terminal. What if there is an error during copy paste? What if someone uses the parameters incorrectly? How do we debug which command was used? Looking at history does not help much.

To make it simple for the team to get on with the task of Python/Spark application execution, we can create a bash shell script as follows:

```
#!/bin/bash
SERVICE_PRINCIPAL=abcd@abcd.com
KEYTAB_PATH=/keytabs/abcd.keytab
MY_JARS=/opt/custom_jars/abcd_1.jar
MAX_RETRIES=128
QUEUE=pyspark
MASTER=yarn
MODE=cluster

spark3-submit --queue ${QUEUE} --conf “spark.yarn.
principal=${SERVICE_PRINCIPAL} --conf “spark.yarn.
keytab=${KEYTAB_PATH} --jars ${MY_JARS} --deploy-mode ${MODE}
--master ${MASTER} $*
```

This demonstrates how powerful a shell script can be and make our life easy. You can try more commands and scripts as per your requirement and explore further. END

By: Bipin Patwardhan

The author is a senior solution architect at an IT MNC in Mumbai. He is working on a project that uses the open source version of Spark along with delta.io for data manipulation and Airflow for workflow management.

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

We Value Your Feedback

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



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

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

OpenSource For You
THE COMPLETE MAGAZINE ON OPEN SOURCE

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

Plotting Data in R: Graphs

R has a number of packages for plotting graphs and data visualisation, such as graphics, lattice, and ggplot2. In this ninth article in the R series, we shall explore the various functions to plot data in R.



We will be using R version 4.1.2 installed on Parabola GNU/Linux-libre (x86-64) for the code snippets.

```
$ R --version
R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 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/.
```

Plot

Consider the all-India consumer price index (CPI - rural/urban) data set up to November 2021 available at *https://data.gov.in/catalog/all-india-consumer-price-index-ruralurban-0* for the different states in India. We can read the data from the downloaded file using the *read.csv* function, as shown below:

```
> cpi <- read.csv(file="CPI.csv", sep=",")

> head(cpi)
Sector Year Name Andhra.Pradesh Arunachal.Pradesh Assam Bihar
1  Rural 2011  January        104            NA   104    NA
2       Urban 2011  January   103            NA   103    NA
3 Rural+Urban 2011  January   103            NA   104    NA
4       Rural 2011 February   107             NA  105    NA
5       Urban 2011 February   106            NA   106    NA
6 Rural+Urban 2011 February   105            NA   105    NA
  Chattisgarh Delhi Goa Gujarat Haryana Himachal.Pradesh
Jharkhand Karnataka
1     105    NA 103    104     104     104      105       104
2     104    NA 103    104     104   103        104       104
3     104    NA 103    104     104              103       105
104
4    107    NA 105     106     106     105    107       106
5    106    NA 105     107     107     105    107       108
6    105    NA 104     105     106     104    106       106
...
```

Let us aggregate the CPI values per year for the state of Punjab, and plot a line chart using the *plot* function, as follows:

```
> punjab <- aggregate(x=cpi$Punjab, by=list(cpi$Year),
FUN=sum)

> head(punjab)
  Group.1       x
1    2011 3881.76
2    2012 4183.30
3    2013 4368.40
4    2014 4455.50
5    2015 4584.30
6    2016 4715.80

> plot(punjab$Group.1, punjab$x, type="l", main="Punjab
Consumer Price Index upto November 2021", xlab="Year",
ylab="Consumer Price Index")
```

The following arguments are supported by the *plot* function:

Argument	Description
x	A vector for the x-axis
y	The vector or list in the y-axis
type	'p' for points, 'l' for lines, 'o' for overplotted plots and lines, 's' for stair steps, 'h' for histogram
xlim	The x limits of the plot
ylim	The y limits of the plot
main	The title of the plot
sub	The subtitle of the plot
xlab	The label for the x-axis
ylab	The label for the y-axis
axes	Logical value to draw the axes

The line chart is shown in Figure 1.

The autocorrelation plot can be used to obtain correlation statistics for time series analysis, and the same can be generated using the *acf* function in R. You can specify the following autocorrelation types: *correlation*, *covariance*, or *partial*. Figure 2 shows the ACF chart that represents the CPI values ('x' in the chart) for the state of Punjab.

The function *acf* accepts the following arguments:

Argument	Description
x	A univariate or multivariate object or vector or matrix
lag.max	The maximum lag to calculate the acf
type	Supported values 'correlation', 'covariance', 'partial'
plot	The acf is plotted if this value is TRUE
i	A set of time difference lags to retain
j	A collection of names or numbers to retain

Bar chart

The *barplot* function is used to draw a bar chart. The chart for Punjab's CPI can be generated as follows, and is shown in Figure 3:

```
> barplot(punjab$x, main="Punjab Consumer Price Index",
sub="Upto November 2021", xlab="Year", ylab="Consumer Price
Index", col="navy")
```

The function is quite flexible and supports the following arguments:

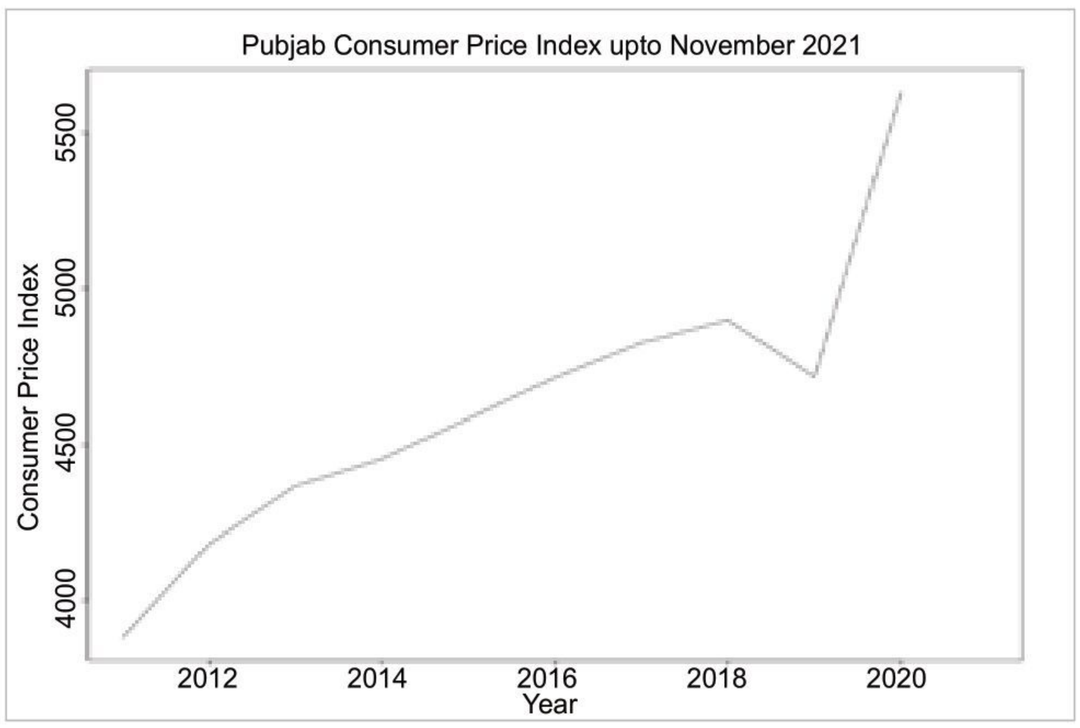


Figure 1: Line chart

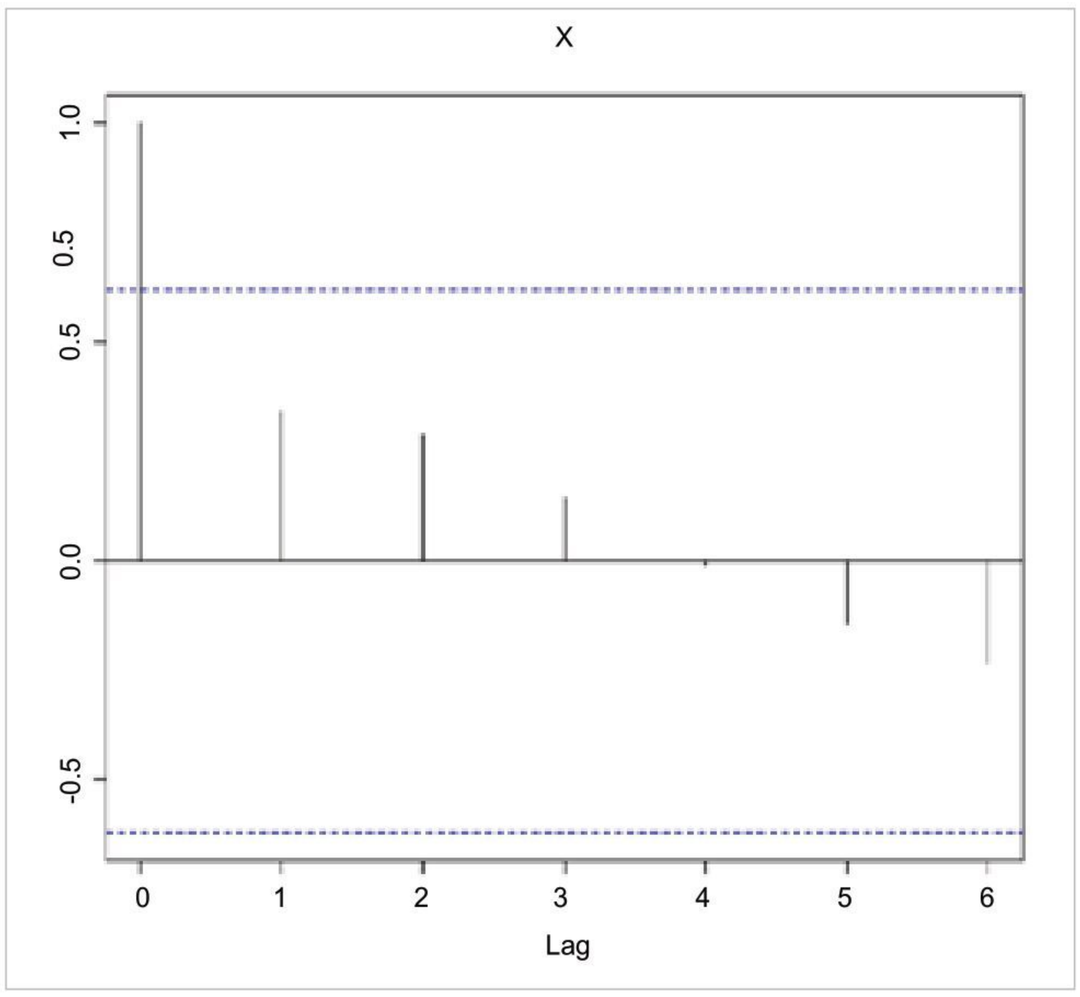


Figure 2: ACF chart

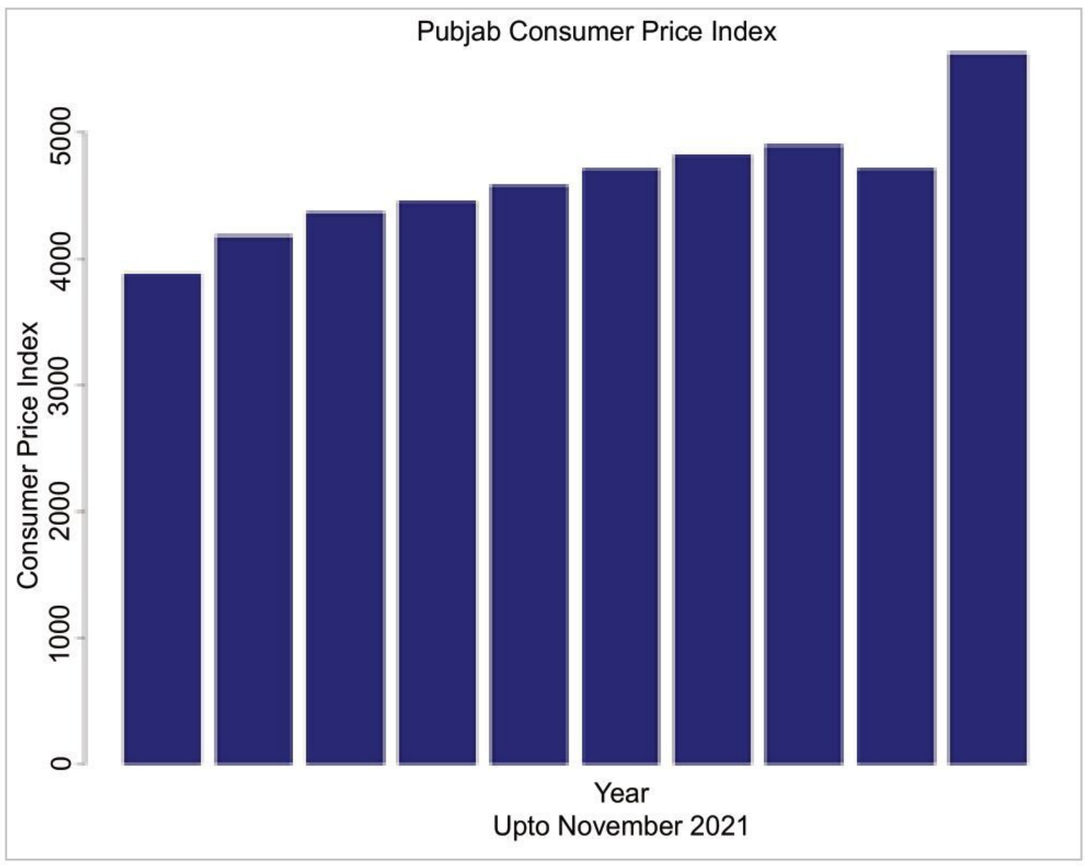


Figure 3: Line chart of Punjab's CPI

Argument	Description
height	A numeric vector or matrix that contains the values
width	A numeric vector that specifies the widths of the bars
space	The amount of space between bars
beside	A logical value to specify if the bars should be stacked or next to each other
density	A numerical value that specifies the density of the shading lines
angle	The angle used to shade the lines
border	The colour of the border
main	The title of the chart
sub	The sub-title of the chart
xlab	The label for the x-axis
ylab	The label for the y-axis
xlim	The limits for the x-axis
ylim	The limits for the y-axis
axes	A value that specifies whether the axes should be drawn

You can get more details on the barplot function using the *help* command, as shown below:

```
> help(barplot)

acf                     package:stats                     R
Documentation

Auto- and Cross- Covariance and -Correlation Function
Estimation

Description:

     The function ‘acf’ computes (and by default plots)
estimates of
     the autocovariance or autocorrelation function.
Function ‘pacf’
     is the function used for the partial autocorrelations.
Function
     ‘ccf’ computes the cross-correlation or cross-
covariance of two
     univariate series.

Usage:

     acf(x, lag.max = NULL,
         type = c(“correlation”, “covariance”, “partial”),
         plot = TRUE, na.action = na.fail, demean = TRUE,
...)

     pacf(x, lag.max, plot, na.action, ...)

     ## Default S3 method:
     pacf(x, lag.max = NULL, plot = TRUE, na.action =
na.fail,
          ...)

     ccf(x, y, lag.max = NULL, type = c(“correlation”,
“covariance”),
         plot = TRUE, na.action = na.fail, ...)

     ## S3 method for class ‘acf’

  x[i, j]
```

Pie chart

Pie charts need to be used wisely, as they may not actually show relative differences among the slices. We can generate the *Rural*, *Urban*, and *Rural+Urban* values for the month of January 2021 for Gujarat as follows, using the *subset* function:

```
> jan2021 <- subset(cpi, Name==”January” & Year==”2021”)

> jan2021$Gujarat
[1] 153.9 151.2 149.1

> names <- c(‘Rural’, ‘Urban’, ‘Rural+Urban’)
```

The *pie* function can be used to generate the actual pie chart for the state of Gujarat, as shown below:

```
> pie(jan2021$Gujarat, names, main=”Gujarat CPI Rural and
Urban Pie Chart”)
```

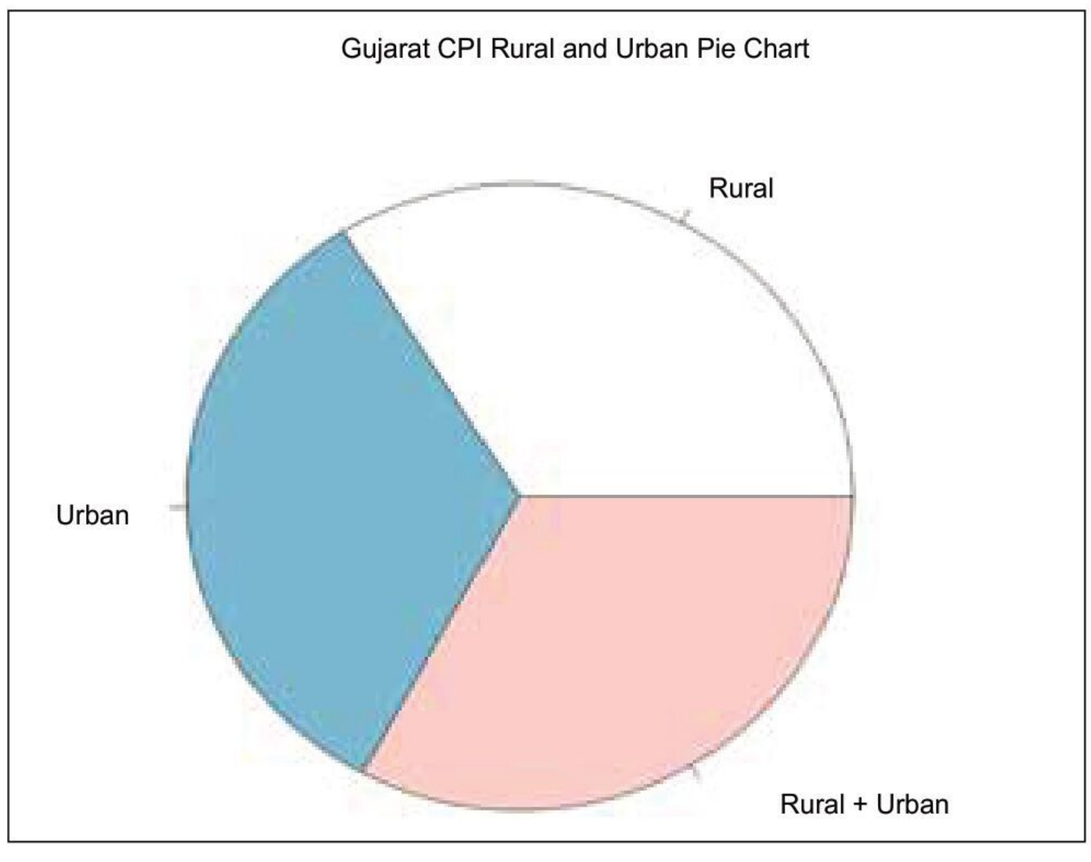


Figure 4: Pie chart

The following arguments are supported by the *pie* function:

Argument	Description
x	Positive numeric values to be plotted
label	A vector of character strings for the labels
radius	The size of the pie
clockwise	A value to indicate if the pie should be drawn clockwise or counter-clockwise
density	A value for the density of shading lines per inch
angle	The angle that specifies the slope of the shading lines in degrees
col	A numeric vector of colours to be used
lty	The line type for each slice
main	The title of the chart

Boxplot

A boxplot shows the interquartile range between the 25th and 75th percentile using two 'whiskers' for the distribution of a variable. The values outside the range are plotted separately. The boxplot functions take the following arguments:

Argument	Description
data	A data frame or list that is defined
x	A vector that contains the values to plot
width	The width of the boxes to be plot-ted
outline	A logical value indicating whether to draw the outliers
names	The names of the labels for each box plot
border	The colour to use for the outline of each box plot
range	A maximum numerical amount the whiskers should extend from the boxes
plot	The boxes are plotted if this value is TRUE
horizontal	A logical value to indicate if the boxes should be drawn horizontally

The boxplot for a few states from the CPI data is shown below:

```
> names <- c ('Andaman and Nicobar', 'Lakshadweep', 'Delhi',
'Goa', 'Gujarat', 'Bihar')
> boxplot(cpi$Andaman.and.Nicobar, cpi$Lakshadweep, cpi$Delhi,
cpi$Goa, cpi$Gujarat, cpi$Bihar, names=names)
```

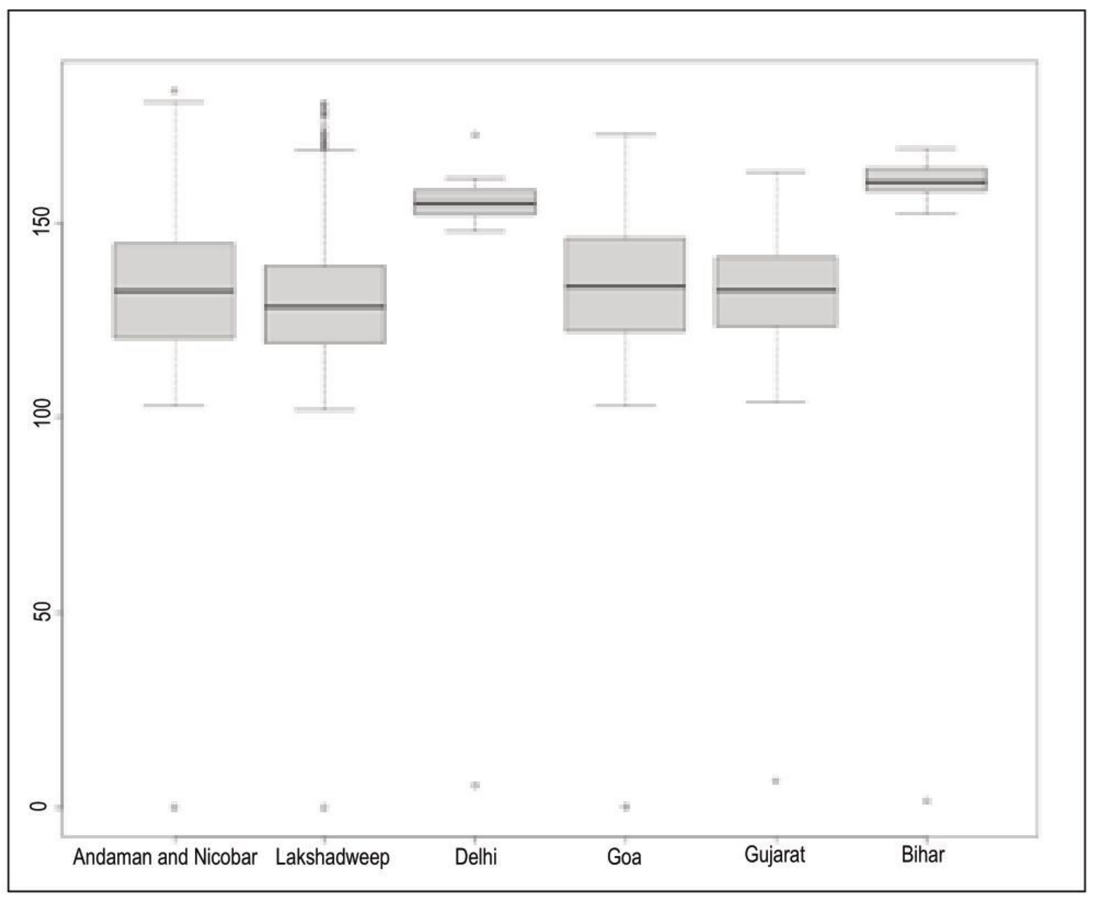


Figure 5: Box plot

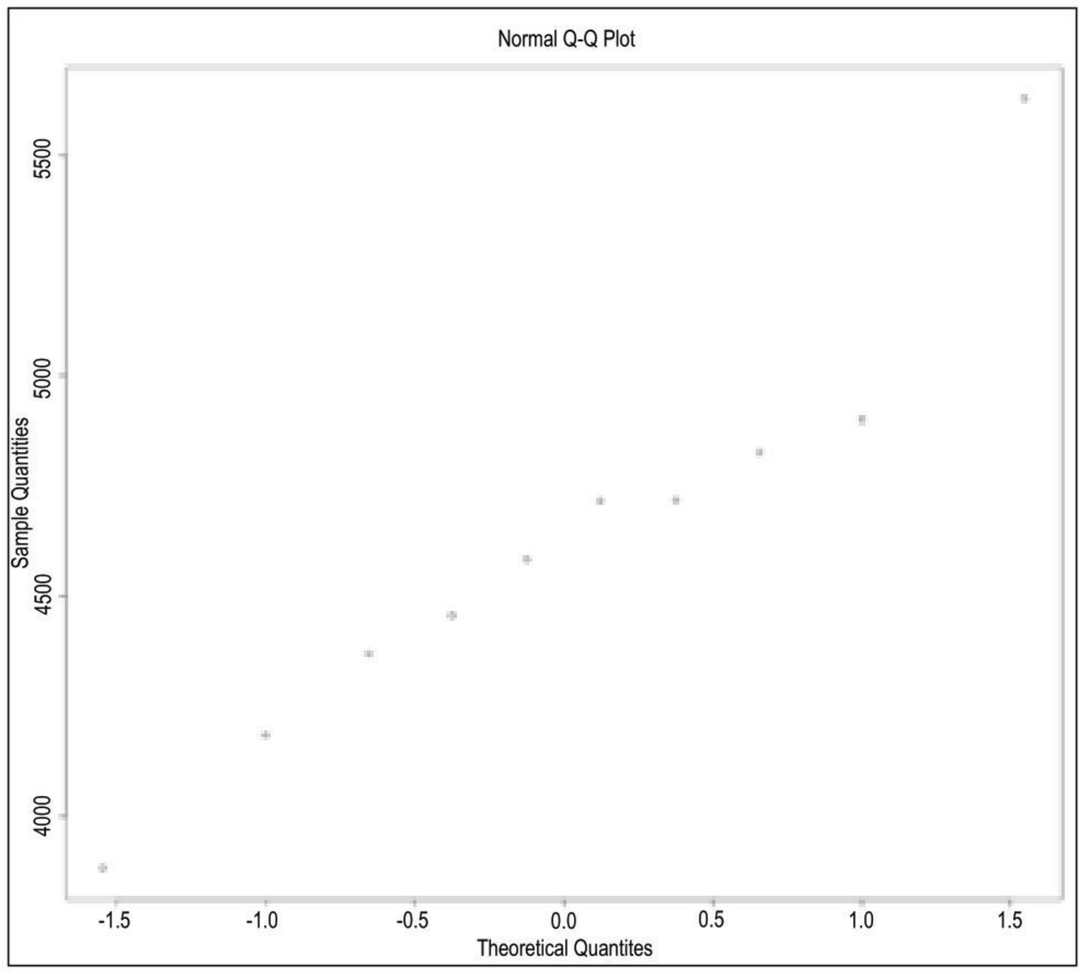


Figure 6: Q-Q plot

Q-Q plot

The Quantile-Quantile (Q-Q) plot is a way to compare two data sets. You can also compare a data set with a theoretical distribution. The *qqnorm* function is a generic function, and we can view the Q-Q plot for the Punjab CPI data as shown below:

```
> qqnorm(punjab$x)
```

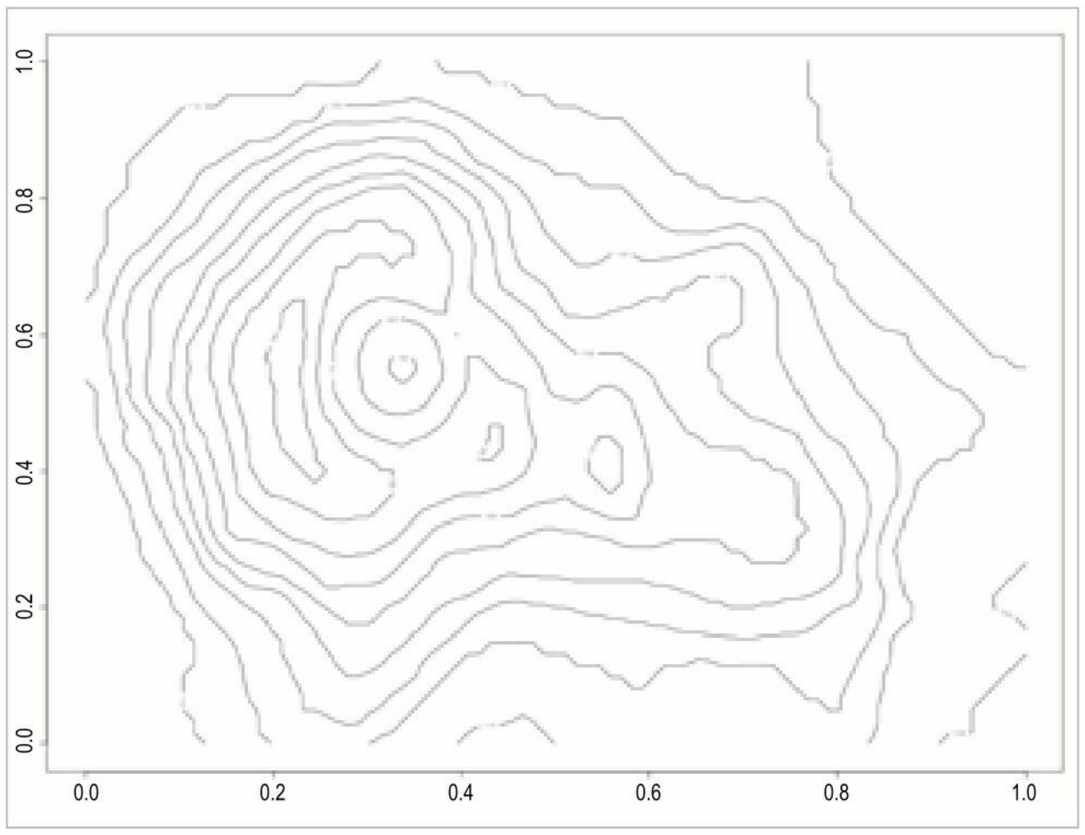


Figure 7: Volcano

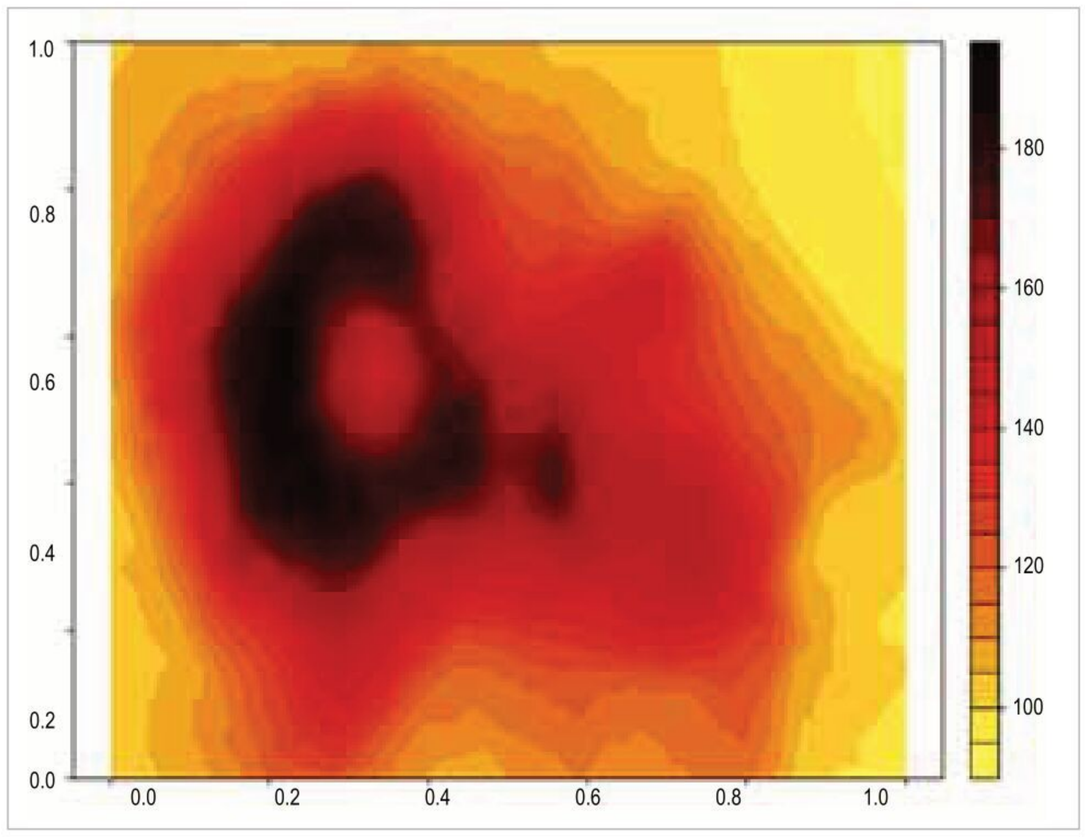


Figure 8: Filled volcano

The *qqline* function adds a theoretical line to a normal, quantile-quantile plot. The following arguments are accepted by these functions:

Argument	Description
x	The first data sample
y	The second data sample
datax	A logical value indicating if values should be on the x-axis
probs	A numerical vector representing probabilities
xlab	The label for x-axis
ylab	The label for y-axis
qtype	The type of quantile computation

Contour plot

The *contour* function is useful for plotting three-dimensional data. You can generate a new contour plot, or add contour lines to an existing chart. These are commonly used along with image charts. The volcano data set in R provides information on the Maunga Whau (Mt Eden) volcanic field, and the same can be visualised with the *contour* function as follows:

```
> contour(volcano)
```

The *contour* function accepts the following arguments:

Argument	Description
x,y	The location of the grid for z
z	A numeric vector to be plotted
nlevels	The number of contour levels
labels	A vector of labels for the contour lines
xlim	The x limits for the plot
ylim	The y limits for the plot
zlim	The z limits for the plot
axes	A value to indicate to print the axes
col	The colour for the contour lines
lty	The line type to draw
lwd	Width for the lines
vfont	The font for the labels

The areas between the contour lines can be filled using a solid colour to indicate the levels, as shown below:

```
> filled.contour(volcano, asp = 1)
```

The same volcano data set with the *filled.contour* colours is illustrated in Figure 8.

You are encouraged to explore the other functions and charts in the graphics package in R. END

By: Shakthi Kannan

The author is a free software enthusiast.

Using a Machine Learning Model to Make Predictions

Machine learning is basically a subset of artificial intelligence that uses previously existing data to make a prediction on new data. Of course, all of us know this by now! This article demonstrates how a machine learning model developed in Python can be used as a part of a Java code to make predictions.



This article assumes you are familiar with the basic development skills and understanding of machine learning. We will start with training our model, and then make a machine learning model in Python.

I am taking the example of a flood prediction model. First, import the following libraries:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Once we have successfully imported the libraries, we need to take in the data sets, as shown in the code below. To predict floods, I am using the river level data set.

```
from google.colab import files
uploaded = files.upload()
for fn in uploaded.keys(): print('User
uploaded file "{name}" with length
{length} bytes'.format(
name=fn, length=len(uploaded[fn])))
Choose files No file chosen
```

The upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable *Saving Hoppers Crossing-Hourly-River-Level.csv to Hoppers Crossing-Hourly-River-Level.csv User uploaded file "Hoppers Crossing-Hourly-River-Level.csv"* with length 2207036 bytes.

Once this is done, we can train our model using the *sklearn library.* For this, we first need to import the library and the algorithm model, as shown in Figure 1.

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

Figure 1: Training the model

```
predicted_riverlevel = regressor.predict(a)
print(predicted_riverlevel)
if (predicted_riverlevel > 1.5):
  print("FLOOD")
else:
  print("No FLOOD")
```

```
[[1.64726206]]
FLOOD
```

Figure 2: Making predictions

```
from sklearn.linear_model import
LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Once that is done we have trained our model, and it's now ready to make predictions, as shown in Figure 2.

Using ML model in Java

What we need to do now is to convert the ML model into a model that can be used by a Java program. There is a library called *sklearn2pmml* that helps us do this:

```
Install the library,
pip install sklearn2pmml
```

Once the library is installed we can convert our already trained model, as shown below:

```
sklearn2pmml(pipeline, 'model.pmml',
with_repr = True)
```

This is it! We can now use the generated *model.pmml* file in our Java code to make predictions. Do try it out! END

By: Jishnu Saurav Mittapalli

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



SQLx:
The Rust SQL Toolkit

An async, pure Rust SQL crate features compile-time checked queries without a domain specific language. This article will give a basic whirlwind tour of the library by showing CRUD operations and the transactions supported using it.

SQLx is a Rust crate for interacting with your database. Some of its features that distinguish it from other crates in the Rust ecosystem are:

- Built with async IO in mind
- Compile-time checked queries
- Database agnostic
- Postgres/MySQL drivers are written in pure Rust (as opposed to them being a FFI binding to a C library)
- Works with various runtimes

Database setup

For our demo, we will start with a simple database with two tables and a simple relationship between them:

```
CREATe TABLE "person"("id" SERIAL8  PRIMARY KEY UNIQUE,"name"
TEXT NOT NULL,"age" smallint NOT NULL,"email" TEXT NULL)
CREATe TABLE "book"("id" SERIAL8  PRIMARY KEY UNIQUE,"name"
TEXT NOT NULL,"isbn" TEXT NOT NULL,"person" INT8 NOT NULL)
ALTER TABLE "book" ADD CONSTRAINT "book_person_fkey" FOREIGN
KEY("person") REFERENCES "person"("id") ON DELETE RESTRICT
ON UPDATE RESTRICT
```

We have two tables:

- Person
- Book

And there is a relationship between the 'book' and the 'person' tables. Make sure you have a local (or remote, if you prefer) PostgreSQL running to test it out. You will also need the *sqlx cli* in your *$PATH*.

Rust project setup

We will use Cargo to create a new project. Cargo is the package manager for Rust. It can download your Rust package's dependencies, compile your packages, make distributable packages, and upload them to *crates.io*. You can refer to Rust documentation for more details.

```
❯ cargo new sqlx-demo
Created binary (application) `sqlx-demo` package
```

The project is set up as a binary project:

```
❯ tree
.
├── Cargo.toml
└── src
    └── main.rs

1 directory, 2 files
```

Let's export the database credentials in an environment variable:

```
export DATABASE_URL=postgres://postgres:postgres@localhost/
sqlx-demo
```

Next, we will add the SQL file *(20211223133946_initial_setup.sql*) as part of the migration:

```
▫ sqlx migrate add initial_setup
Creating migrations/20211223133946_initial_setup.sql
```

Congratulations on creating your first migration!
Did you know you can embed your migrations in your application binary? On startup, after creating your database connection or pool, add:

```
sqlx::migrate!().run(<&your_pool OR &mut your_connection>).
await?;
```

Note that the compiler won't pick up new migrations if no Rust source files have changed. You can create a Cargo build script to work around this with `sqlx migrate build-script` (see *https://docs.rs/sqlx/0.5/sqlx/macro.migrate.html*).

Now you have to go to the above file and add your SQL statements. Once you have done that, you can set up the database:

```
▫ sqlx database setup
Applied 20211223133946/migrate initial setup (27.149609ms)
```

Next, let's add the required dependencies to the *Cargo.toml* file:

```
[dependencies]
anyhow = "1.0.51"
sqlx = { version = "0.5.9", features = [  "runtime-tokio-
rustls", "postgres" ]}
tokio = { version = "1", features = ["full"] }
```

And you can modify the *main.rs* code to do the following things:

- Set up a connection pool
- Pass credentials to get a connection from the pool

```
use sqlx::{PgPool};
use anyhow::*;
use sqlx::postgres::PgPoolOptions;

#[derive(Clone)]
pub struct DBApplication {
    pool: PgPool
}

impl DBApplication {
    pub async fn new(config: String) -> Result<DBApplication> {
        let pool = PgPoolOptions::new()
            .max_connections(5)
            .connect(&config)
            .await?;
        Ok(DBApplication { pool })
    }
}

#[tokio::main]
async fn main() -> Result<()> {
    let db = DBApplication::new("postgres://postgres:postgres@
localhost/sqlx-demo".into()).await?;
    println!("Connection acquired!");
    Ok(())
}
```

You can confirm if this works by running the command *cargo run:*

```
▫ cargo run
    Finished dev [unoptimized + debuginfo] target(s) in 0.04s
     Running `target/debug/sqlx-demo`
Connection acquired!
```

Domain types

We have two tables and, usually, we would want to model them into Rust structs. The easy way of doing this is to use *derive* macros:

```
#[derive(sqlx::FromRow, Debug)]
pub struct Person {
    id: i64,
    name: String,
    age: i16,
    email: String
}
```

The usual convention is to have two sets of records -- one for inserting and the other for fetching. I use the *New* prefix to distinguish them:

```
pub struct NewPerson {
    pub name: String,
    pub age: i16,
    pub email: String
}
```

Another nice option is to use the *query_as!* macro, as it doesn't need the above *derive* macro. I will be using that later in the demo.

Transactions

Let's write a function to create a person, and let's use a transaction to do the same:

```
pub async fn create_person(&self, person: NewPerson) ->
Result<Person> {
    let mut transaction = self.pool.begin().await?;
    let person = sqlx::query_as("INSERT INTO person(name,
age, email) VALUES ($1, $2, $3) RETURNING *;")
        .bind(person.name)
        .bind(person.age)
        .bind(person.email)
```

```
        .fetch_one(&mut transaction).await?;
    transaction.commit().await?;
    Ok(person)
}
```

You can modify your main fetch to use it:

```
#[tokio::main]
async fn main() -> Result<()> {
    let db = DBApplication::new("postgres://
postgres:postgres@localhost/sqlx-demo".into()).await?;
    let sibi = db.create_person(NewPerson{ name: "Sibi".
into(), age: 20, email: "test@psibi.in".into()}).await?;
    println!("Sibi: {:?}", sibi);
    Ok(())
}
```

Now let's check if the function returns early because of an error, if the transaction rolls back. We will simulate a failure in the above function by an early return after insert:

```
pub async fn create_person(&self, person: NewPerson) ->
Result<Person> {
    let mut transaction = self.pool.begin().await?;
    let person = sqlx::query_as("INSERT INTO person(name,
age, email) VALUES ($1, $2, $3) RETURNING *;")
        .bind(person.name)
        .bind(person.age)
        .bind(person.email)
        .fetch_one(&mut transaction).await?;
    bail!("Simulate error");
    transaction.commit().await?;
    Ok(person)
}
```

And the main function will be like this:

```
#[tokio::main]
async fn main() -> Result<()> {
    let db = DBApplication::new("postgres://
postgres:postgres@localhost/sqlx-demo".into()).await?;
    let sibi = db.create_person(NewPerson{ name: "Sibi".
into(), age: 20, email: "test@psibi.in".into()}).await?;
    println!("Sibi: {:?}", sibi);
    Ok(())
}
```

Before running the executable, you can check the current rows in your table:

```
sqlx-demo=# select * from person;
 id | name | age | email
----+------+-----+-------
(0 rows)
```

Now let's execute the program:

```
▫ cargo run     Finished dev [unoptimized + debuginfo]
target(s) in 0.05s
     Running `target/debug/sqlx-demo`
Error: Simulate error
```

You can see that it did return an error. You can also verify that the transaction has rolled back and has not inserted the row, by inspecting the rows:

```
sqlx-demo=# select * from person;
 id | name | age | email
----+------+-----+-------
(0 rows)
```

Other CRUD operations

Similarly, you can implement other CRUD (create, read, update, delete) operations too:

```
pub async fn get_person(&self, id: i64) -> Result<Person> {
    let mut transaction = self.pool.begin().await?;
    let person = sqlx::query_as("SELECT * from person where
id = $1")
        .bind(id)
        .fetch_one(&mut transaction).await?;
    transaction.commit().await?;
    Ok(person)
}

pub async fn update_person_name(&self, id: i64, name: String)
-> Result<Person> {
    let mut transaction = self.pool.begin().await?;
    let person = sqlx::query_as("UPDATE person SET name = $1
WHERE id = $2 RETURNING *")
        .bind(name)
        .bind(id)
        .fetch_one(&mut transaction).await?;
    transaction.commit().await?;
    Ok(person)
}

pub async fn delete_book(&self, id: i64) -> Result<()> {
    let mut transaction = self.pool.begin().await?;
    sqlx::query("DELETE FROM book WHERE id = $1")
```

```
        .bind(id)
        .fetch_optional(&mut transaction).await?;
    transaction.commit().await?;
    Ok(())
}
```

Compile time queries

One nice thing about SQLx is that it can perform static checks against your database server. Using the *query_as* macro will achieve this. Rewriting the above *get_person* function to use the macro will look like this:

```
pub async fn get_person_macro(&self, id: i64) ->
Result<Person> {
    let mut transaction = self.pool.begin().await?;
    let person = sqlx::query_as!(Person, "SELECT * from
person where id = $1", id)
        .fetch_one(&mut transaction)
        .await?;
    transaction.commit().await?;
    Ok(person)
}
```

The real power of it shines when you make mistakes in your SQL query. Let's change the above code to use *id_wrong* instead of *id:*

```
pub async fn get_person_macro(&self, id: i64) ->
Result<Person> {
    let mut transaction = self.pool.begin().await?;
    let person = sqlx::query_as!(Person, "SELECT * from
person where id_wrong = $1", id)
        .fetch_one(&mut transaction)
        .await?;
    transaction.commit().await?;
    Ok(person)
}
```

Compiling the above code will give this error:

```
error: error returned from database: column "id_wrong" does
not exist
  --> src/main.rs:99:22
   |
99 |         let person = sqlx::query_as!(Person, "SELECT *
from person where id_wrong = $1", id)
   |                      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
   |
   = note: this error originates in the macro `$crate::sqlx_
macros::expand_query` (in Nightly builds, run with -Z macro-
backtrace for more info)
```

As the error message indicates, there is no column named *id_wrong* in your table.

Migrations

You could see above that we ran the migrations using the *sqlx-cli* tool. But in our application code, it will often be handy to run the migrations on a fresh database. The *sqlx::migration* macro is specifically meant for that. It embeds the migration into your binary and can be used to run it before the application is started. This is how your new main function will look like with migrations integration:

```
#[tokio::main]
async fn main() -> Result<()> {
    let db = DBApplication::new("postgres://
postgres:postgres@localhost/sqlx-demo".into()).await?;
    sqlx::migrate!().run(&db.pool).await?;
    ...
    ...
    Ok(())
}
```

Hopefully, this article gave you a good idea of how to use SQLx for interacting with your database. If you prefer an object-relational mapping (ORM) based solution, these are some of the popular options:

- Diesel: *https://diesel.rs/*
- sea_orm: *https://docs.rs/sea-orm/latest/sea_orm/*
- ormx: *https://github.com/NyxCode/ormx*

Both sea_orm and ormx are built on top of SQLx. The major differentiating factor between Diesel and the other ORM based solutions is that the former isn't async IO ready yet. END

References

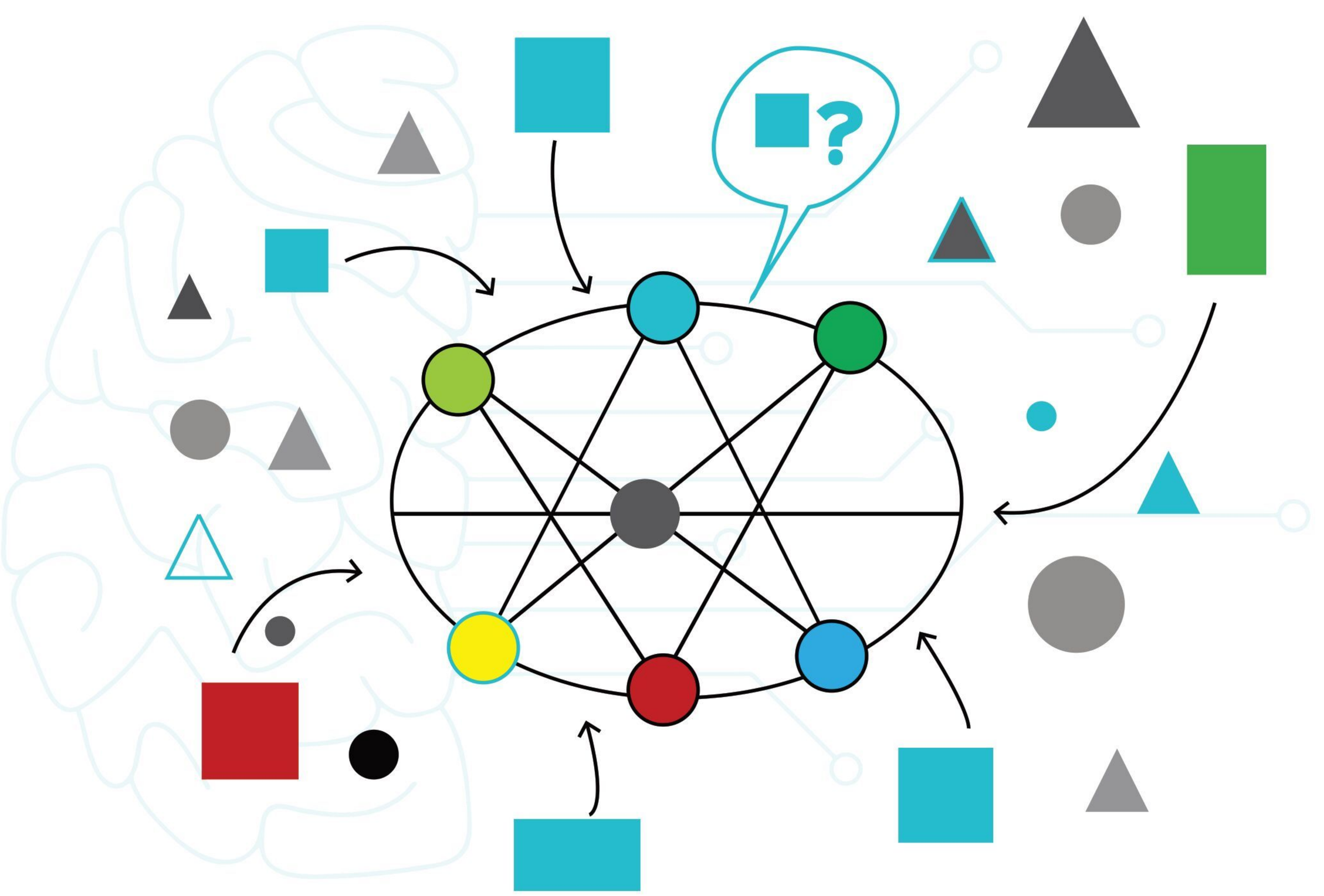
[1] GitHub home page: *https://github.com/launchbadge/sqlx*
[2] Field attributes documentation: *https://docs.rs/sqlx/latest/sqlx/trait.FromRow.html*
[3] Rust types reference: *https://docs.rs/sqlx/latest/sqlx/postgres/types/index.html*

By: Sibi Prabakaran

The author is a senior engineer working at FPComplete.

What is Federated Learning?

Federated learning is a distributed machine learning approach for developing and training models on a global scale. This article introduces a few open source frameworks that help to explore it, which every newbie must know of.



Federated learning helps to develop and train models for distributed machine learning. Loosely translated, federation means a group of objects working towards the same common goal. In federated learning, the model is developed with the help of updates from a large group of devices, usually phones. Prior to the advent of federated learning, devices leveraging various artificial intelligence solutions sent repeated messages embedded with the data stored on the device. This led to two major bottlenecks:

- *Communication:* Not all devices in question had constant Internet access. Moreover, the bandwidth was very often insufficient for such a transfer.
- *Privacy:* Any third party actor with a malicious intent could figure out a way to access the data coming in from a consumer.

To address all of this and much more, Google introduced federated learning. Originally built into the Google keyboard, it quickly piqued the interest of the research community. The benefits of federated learning are:

- Leverages the on-device compute power of modern-day electronics
- Protects the privacy of the customer with zero data transfers
- Can function in a decentralised architecture to handle single-point failures
- Trains over a high number of clients even with slow or low-quality Internet

- Is robust and can handle client dropouts up to a threshold value

Although federated learning was developed and visualised as an enterprise solution, there are quite a few open source frameworks that help you explore it even if you have access to a single desktop. Let's take a quick look at them.

TensorFlow Federated

TensorFlow is one of the most popular frameworks used for building machine learning and deep learning models. Its popularity is nestled in the ability to run TensorFlow models reliably on any device — from a custom server to a handheld IoT device — and its ease of integration and extensive support for tooling. TensorFlow Federated is an open source framework built to support research and development of machine learning models using federated learning on decentralised data. It allows developers and researchers to simulate existing model architectures as well as test novel architectures for federated learning. It has two main interfaces.

Federated Core API: This is a programming environment for developing distributed computations. It serves as a foundation for the Federated Learning API and is used by systems researchers

Federated Learning API: This high-level interface helps machine learning developers and researchers incorporate federated learning without having to look into the low-level details.

Further documentation for this framework is available at *https://www.tensorflow.org/federated*.

PySyft + PyGrid

PySyft is a Python library that aids in developing federated learning models for the purpose of research. It uses differential privacy and encrypted communications. It works in tandem with existing deep learning frameworks such as TensorFlow and PyTorch. However, PySyft alone isn't sufficient to work on problems that involve communications over a network. This is where PyGrid comes in, as it helps to implement federated learning on a wide variety of devices and to deploy PySyft at scale.

Further documentation is available at *https://github.com/OpenMined/PySyft*.

OpenFL

Developed by Intel Labs, OpenFL is a federated learning framework built to be flexible, extensible, and learnable. It splits nodes into collaborators and aggregators. Collaborators train the model on their local data and send updates. Aggregators use updates from collaborators to update the global model. Communication is done between nodes using mTLS, which warrants that each node holds a valid certificate. OpenFL allows customisations in the form of lossy or lossless compression, logging, data split and a variety of aggregation logic.

For further documentation on this framework, you can refer to *https://github.com/intel/openfl*.

APPFL: Argonne Privacy-Preserving Federated Learning

APPFL is an open source framework that helps develop novel models aimed towards privacy-preserving federated learning. It provides the framework for users to develop neural networks on decentralised data with differential privacy. Apart from standard implementations and data sets, it allows users to build customised models using the PyTorch module. Models developed with APPFL can be run from the terminal, using notebooks or on the Google Cloud platform. A developer can also simulate runs on a single or clustered environment with serial or parallel execution.

For the complete documentation, you can go to *https://github.com/appfl/appfl*.

Flower

Developed by Adap, Flower is a unified approach to federated learning, analytics and evaluation. It was developed to be scalable to work with a wide range of machine learning frameworks and devices. Flower is scalable to the tune of handling tens of millions of clients. It works efficiently with a multitude of deep learning frameworks such as Keras, TensorFlow, PyTorch, Numpy, and so on. It was developed to run with the same efficiency on all kinds of devices such as servers, laptops, mobiles and other IoT and edge devices. It helps researchers deploy their novel algorithms with ease.

Further documentation on this framework is available at *https://flower.dev/*.

Since its inception, researchers and machine learning developers have been experimenting with federated learning on a small scale with very minimal upfront infrastructure. The frameworks introduced in this article are some of the most well-documented and beginner-friendly solutions for understanding and implementing federated learning. END

By: Aanchal Narendran

The author is researcher at Centre for Cloud Computing and Big data.

Creating a GitHub Action from Scratch

DevOps practices use automation to carry out repetitive tasks, leaving you free to focus on the more challenging but fun parts of building software. This article will explain how to use GitHub Actions, the automation platform baked into all GitHub projects for free.



GitHub Actions makes it easy to automate all the software workflows that you see in your program — from code reviews, branch management and issue triaging, to even a complete CI/CD pipeline.

GitHub Actions runs workflows when other events happen in your repository, such as when code is pushed, a pull request is opened or an issue is created. 'Workflows' are YAML specs that are defined within the *.github/workflows* directory in your repository. These workflows contain one or more 'jobs' that can be run sequentially or in parallel. Each job will run inside its own virtual machine 'runner' or inside a container. Each job has one or more steps that either run a script you define, or run an 'action' which is a reusable extension that can simplify your workflow. These actions can be either those that you create yourself, or use for free from the GitHub Marketplace. Figure 1 illustrates how all of this works together in accomplishing the automation that you need.

Let's understand this better through an example that many of us might have seen in our CI pipelines — when code is pushed, create a build (Figure 2).

In this example, the 'event' that happens in your repository is a 'push' event, and you need automation to create a build of your code. To accomplish this, you need to create a workflow YAML file within the *.github/workflows* directory, and define all the necessary components of jobs, runners and steps necessary for your build to take place.

Workflow files

Let's look at how to write workflows in YAML to define the automation that you need. Figure 3 gives an example of what a workflow file will look like, along with all the elements necessary.

In this example, you can see the 'on' key that defines the event in your repository. On clicking this, the workflow is triggered to run automatically using GitHub Actions. Your workflow can have multiple triggers, and even add qualifiers to

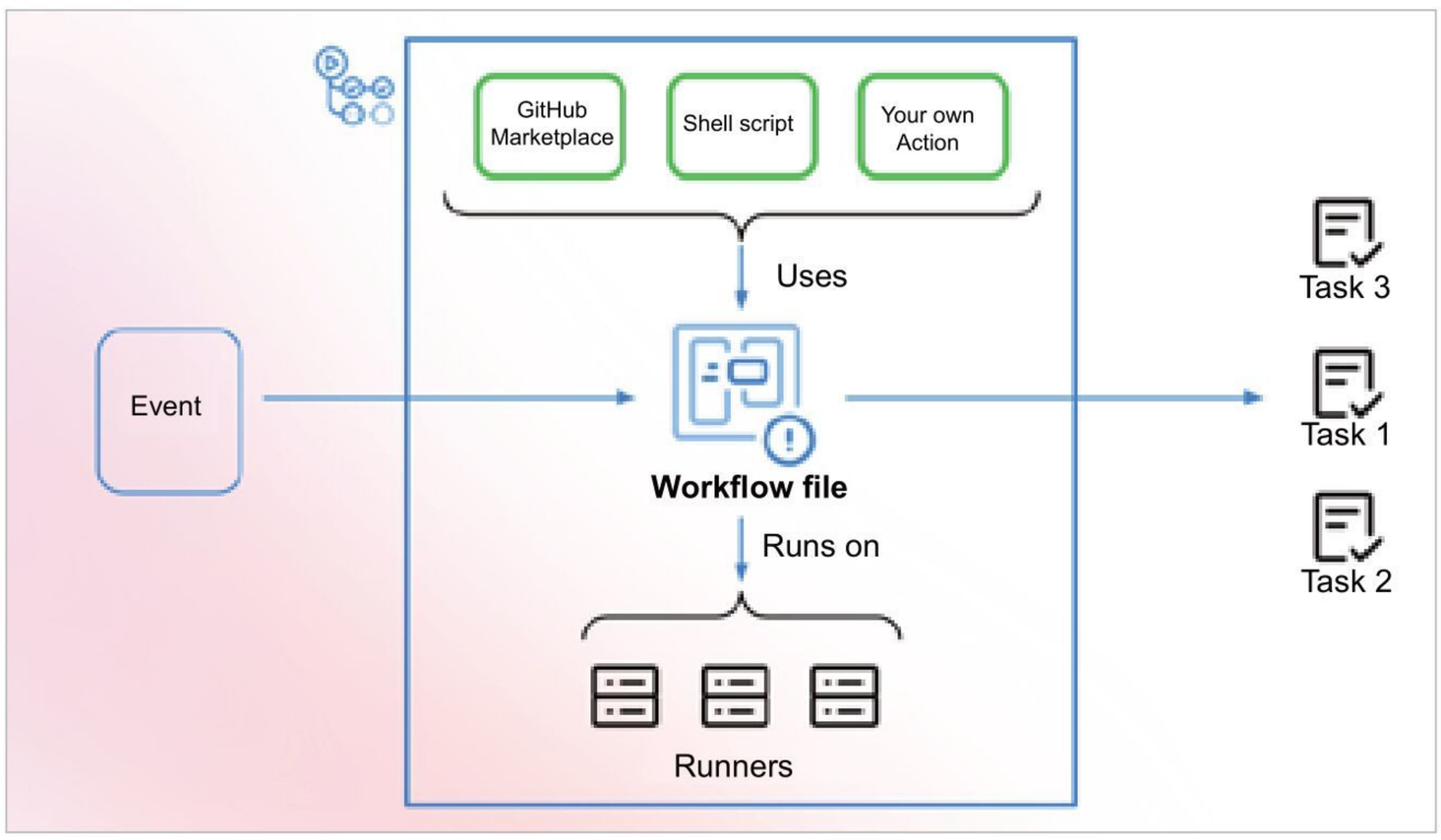


Figure 1: How GitHub Actions work

certain events (like triggering a workflow only when a push event occurs on the main branch).

Followed by this is a 'jobs' key that defines all the jobs in the workflow, with a name for each job. In our example, there is only one job named 'build' that runs on its own virtual machine runner with the *ubuntu-latest* image as defined by the *runs-on* key.

Within the job is a list of steps that need to run in sequence on the same virtual machine in order to accomplish the task of the job. In our example, these might be to do whatever is required in order to create a build.

In Figure 3, the first step uses an 'action' from the GitHub Marketplace that does a check of the code; the next two steps run scripts within the shell of the virtual machine runner, and the last step runs a custom action that's defined in the repository. You can use any combination of these scripts and actions necessary to accomplish the tasks for your job.

Why use GitHub Actions?

GitHub Actions is native to GitHub and integrates well with the common software development workflows you might be familiar with on it. You can accomplish many of your automation needs solely with GitHub Actions, or even combine them with other tools to manage your workflows in a seamless way.

Actions is independent of the language/framework that you use for your project, and you can use it to automate any of your workflows. Whether you want to run simple linting using JS libraries, create a distributable from your code, build a container image, or do anything else in between, you can accomplish it using GitHub Actions.

Since workflow jobs run on virtual machine runners, you can run your workflows on Linux, macOS, Windows or even ARM. You can choose which architecture and OS you want for each of your jobs within your workflow, based on your own use case. These virtual machine runners can be hosted by GitHub, or you can even self-host them. Alternatively, you can choose to skip the virtual machine runner altogether and just run a container image when your workflow gets triggered.



Figure 2: Example of push and build

```yaml
name: CI

on:
  push:
    branches: [ main ]
  pull_request:
    branches: [ main ]
  workflow_dispatch:

jobs:
  build:
    runs-on: ubuntu-latest

    steps:

    - name: Using the Checkout action from Marketplace
      uses: actions/checkout@v2

    - name: Run a one-line script
      run: echo Hello, world!

    - name: Run a multi-line script
      run: |
        echo Add other actions to build,
        echo test, and deploy your project.

    - name: My own action in the same repo
      uses: ./
```

Figure 3: Example workflow file

GitHub Actions is used by many developers, open source projects and businesses for their automation needs. If you check out various open source projects on GitHub, you might find that many of them are using Actions for their project management, CI/CD pipelines or other automations. There are more than 12,000 community-powered Actions extensions on the



Figure 4: Components of your own GitHub Action

GitHub Marketplace that you can use readily within your own workflows and build on top of.

Lastly, GitHub Actions is completely free for public repositories, and has free limits for private repositories as well.

Why build your own action?

While there are thousands of actions available on the GitHub Marketplace that you could use, there could be scenarios where you might have to create your own action extension for use within your workflows.

First, if you want to perform certain tasks or have some custom logic for which an action extension doesn't already exist on the Marketplace, it is often useful to write your own action. This helps make your workflow modular, and also enables reuse and sharing within your teams, projects or organisations.

Second, if you want to integrate GitHub and any other services/tools that you might use, you can do this easily by creating a GitHub Action that interacts with the APIs of those services/tools. This makes it easier for you to interact with other tools from within your workflows through GitHub Actions.

One of the main benefits of GitHub Actions being native to GitHub is that it is accessible to the millions of developers using the latter. If an action that you create on your own can be used by other developers as well, you can publish it yourself for free on the GitHub Marketplace and make it available to everyone on GitHub.

How to build your own action

An action has essentially two components (Figure 4). One is a YAML file called *action.yml* and the other is the source code of your action.

Let us look at what exactly these two components do.

The *action.yml* file stores the metadata about the action you are building. It stores details like the name of the action, what are the inputs it needs and the outputs it delivers. It also defines whether the source code of the action is in JavaScript that can be run using Node12, or is a Docker image. It also contains information about where to find the JavaScript files or the Docker image to run the action.

The source code part of your action contains the actual logic of it. This can be JavaScript files that can run using Node12, a Dockerfile that you are going to build when an action runs, or it can simply be a Docker image that you refer to. It will have access to the event payload that triggered the workflow and also the context of the workflow run through environment variables (ENV). You can easily call GitHub's APIs or other APIs from within your source code as a part of your logic, to accomplish the action's task.

GitHub Actions lets you create your own actions using the millions of open source libraries accessible on GitHub, and allows you to write them in JavaScript or create a container action. It makes software workflow automation easy, which helps you be more productive and eases your development life cycle.

You can automate various use cases by using GitHub Actions, including API management, code quality, support, chat, code review, publishing, deployment, localisation, continuous integration, learning, project management, monitoring, security, dependency management, and others. END

References

[1] Understanding GitHub Actions: *https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions*
[2] Creating custom actions: *https://docs.github.com/en/actions/creating-actions/about-custom-actions*
[3] Actions on the GitHub Marketplace: *https://github.com/marketplace?type=actions*
[4] Interactive learning on GitHub with Learning Lab: *https://lab.github.com/*

By: M.V. Karan

The author is a senior developer relations manager at GitHub. This article is based on a tech talk he gave at OSI 2021 (virtual).

Transcribed and curated by Laveesh Kocher.

Simplifying Cloud Native Development with Skaffold



The successful rollout of an application entails several essential steps. If the developer misses any of these in the development stage, it could make the application susceptible to problems later, leading to rework and wastage of the developer's time. Skaffold automates these steps and assists developers in a hassle-free rollout of their applications.

In a developer's workflow, a wide variety of changes are first made to the code. Then container images are built, which are pushed to a registry till the application is eventually deployed. This entire process can entail a lot of inefficiencies. This is where Skaffold comes in – it helps to manage the developer workflow.

Understanding the developer workflow

This is what the developer workflow looks like.

- ***Code changes:*** The first step simply involves changing the code. While you are working on your code, you will simply make changes to it as per your own requirements and save it.
- ***Containerise:*** Once you are done making changes to your code, you will containerise it to see it actually work. You may use Docker and a *build* command to make a sample Docker file, then use that file to build the Docker image, and tag that image with a specific version.
- ***Push:*** After you have containerised the image, you are supposed to push it into a registry.
- ***Deploy:*** If you have got a Kubernetes deployment YAML, you can apply it to the particular environment of the Kubernetes cluster that you are pointing to, using the *kubectl* command that you can see in Figure 1. Once that is done, your application will be deployed.
- ***Connect and view logs:*** The next thing is to check if everything is working or not. So you will connect

to that particular cluster, check the pod forwarding, and then stream the logs to see if everything is working well or not.

All in all, you will have to keep repeating the same steps as per the number of changes you make and the number of changes you want to look at. So we now need a tool like Skaffold that can detect the activities that are common in nature, and automate and take care of these behind the scenes. Let us understand this from a developer workflow perspective.

Automating common activities

In Figure 2, you can see some key activities marked in green that developers care about – things like coding and configuring the application to make it run. Marked in red are the things that they hope will just get done automatically by a tool to avoid the repetitive work that they are doing as part of the developer workflow. Skaffold allows developers to work on what they care about and takes care of the rest of the things.

Skaffold's end-to-end development pipelines are as follows.

- *$ skaffold dev:* Whenever you make a change to your code, the *dev* cycle automatically gets activated.
- *$ skaffold run: Run* allows you to run your particular application.
- *$ skaffold debug:* As the name suggests, *debug* helps in debugging a particular application that you want as a part of this process.

Once Skaffold has sensed the changes being made to the code and to the application's configuration, it will perform all the steps — building the code, filling the containers, pushing it to the registry, applying any configuration deployment and bringing the pods up on its own. It will then stream the logs to you so you can actually see that. It is capable of working with various clusters and registries.

```
$ docker build . -t gcr.io/project-id/myapp:v2
$ docker push gcr.io/project-id/myapp:v2
$ vi myapp-deployment.yaml
$ kubectl apply -f myapp-deployment.yaml
$ kubectl get pods
$ kubectl port-forward some-pod-qx54dw xxxx:yyyy
$ kubectl logs -f some-pod-qx54dw
```

Figure 1: *kubectl* command is initiated

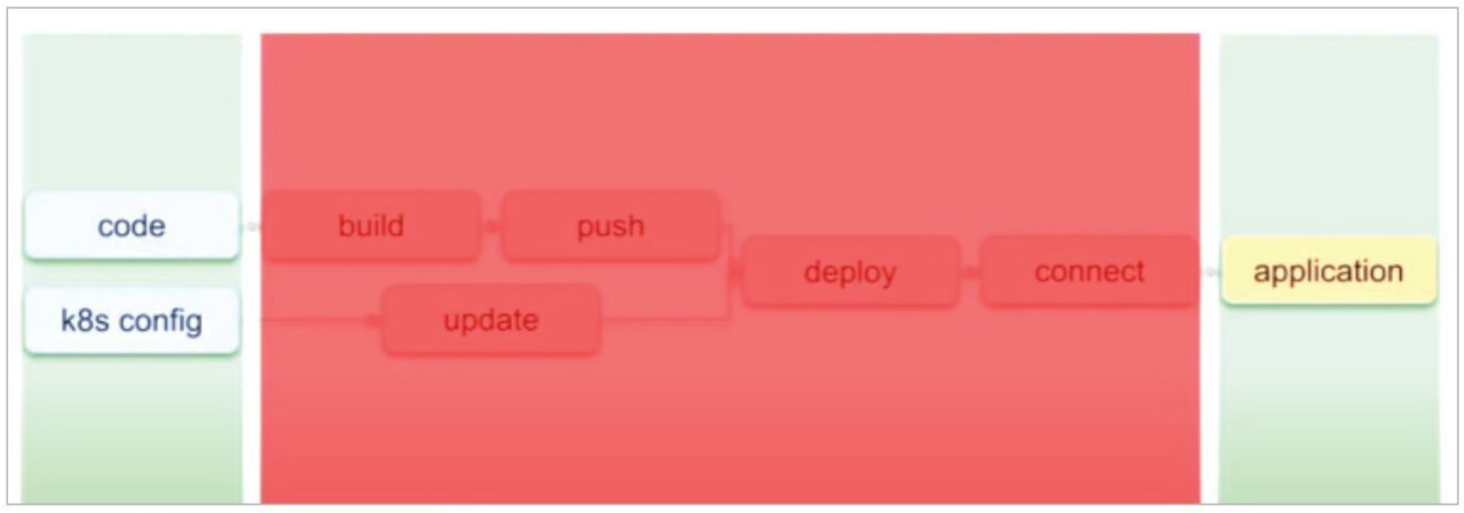


Figure 2: Steps in the development process

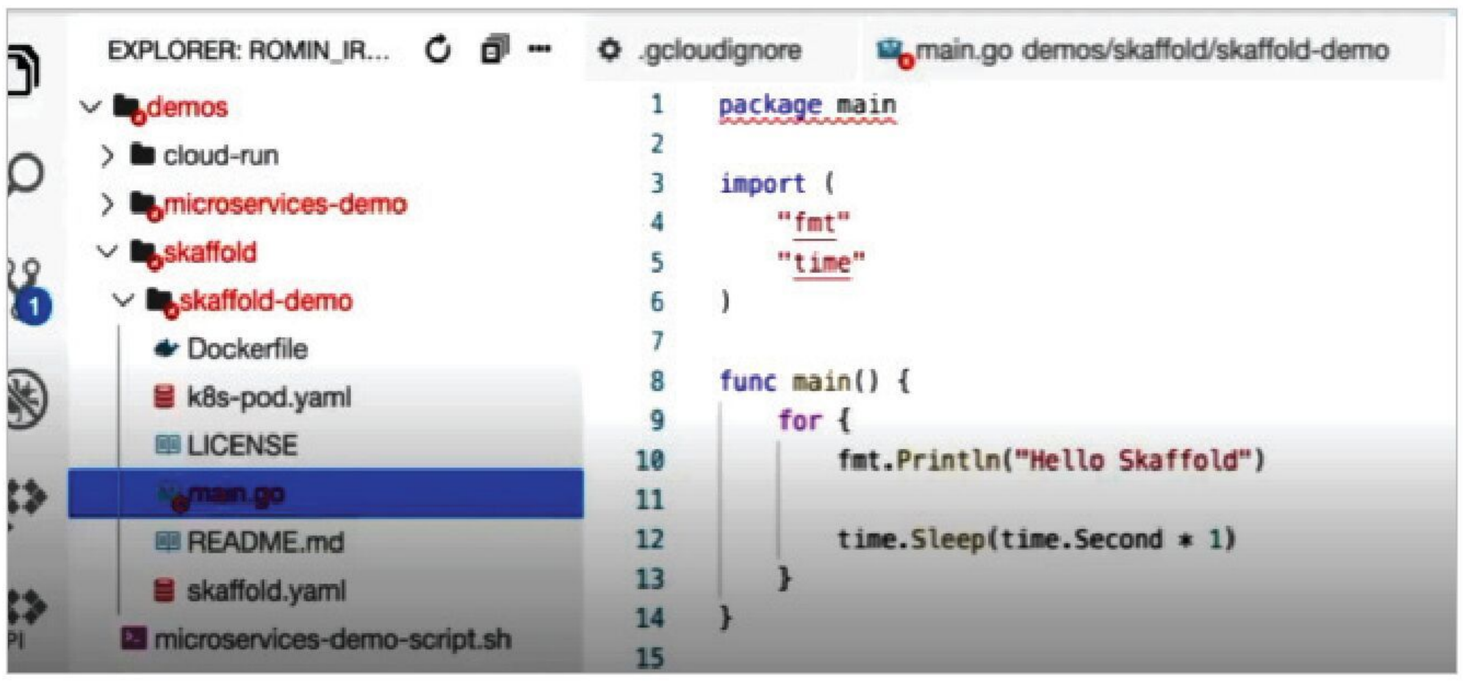


Figure 3: main.go

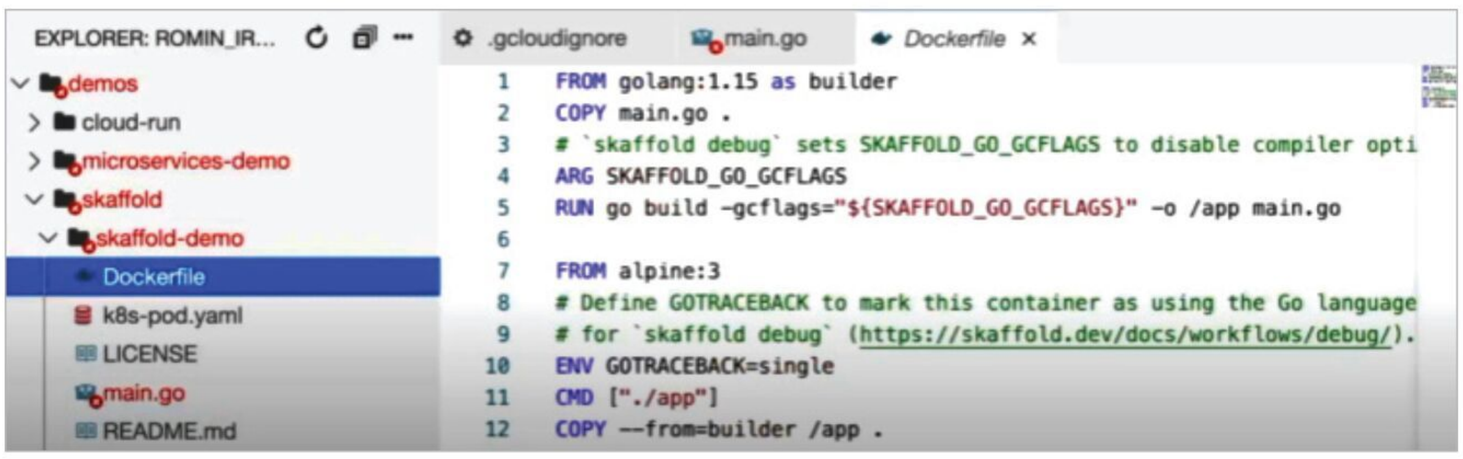


Figure 4: Folder consisting of Dockerfile

A demo

Here we have got a very straightforward *go* file (Figure 3). This file contains one main file and one main function, and it is printing out 'Hello Skaffold'.

In the same folder, we have got a Dockerfile that you can see in Figure 4. This is a multi-staged Dockerfile. We are essentially just creating an executable for this file in the app, and in the next step we are going to set the default command to run when the container is up, which will in turn generate the output file. This is from

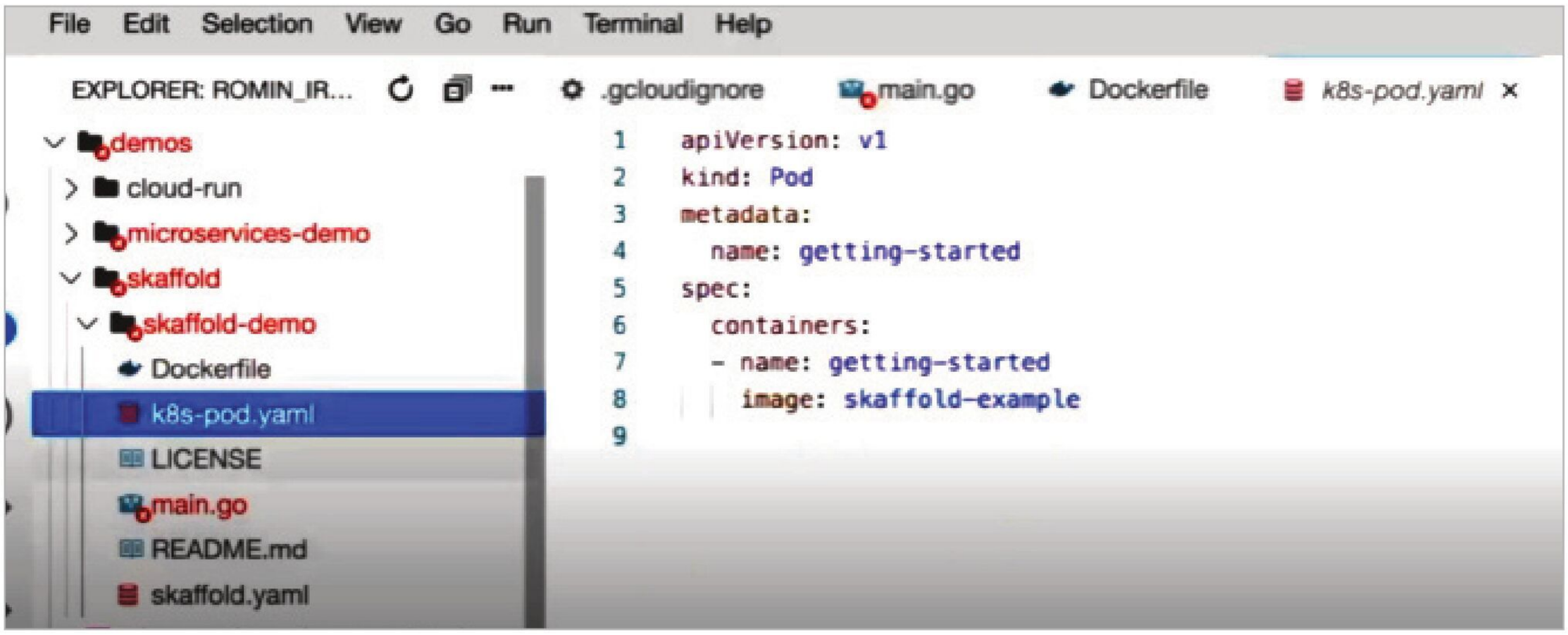


Figure 5: *k8s-pod.yaml*

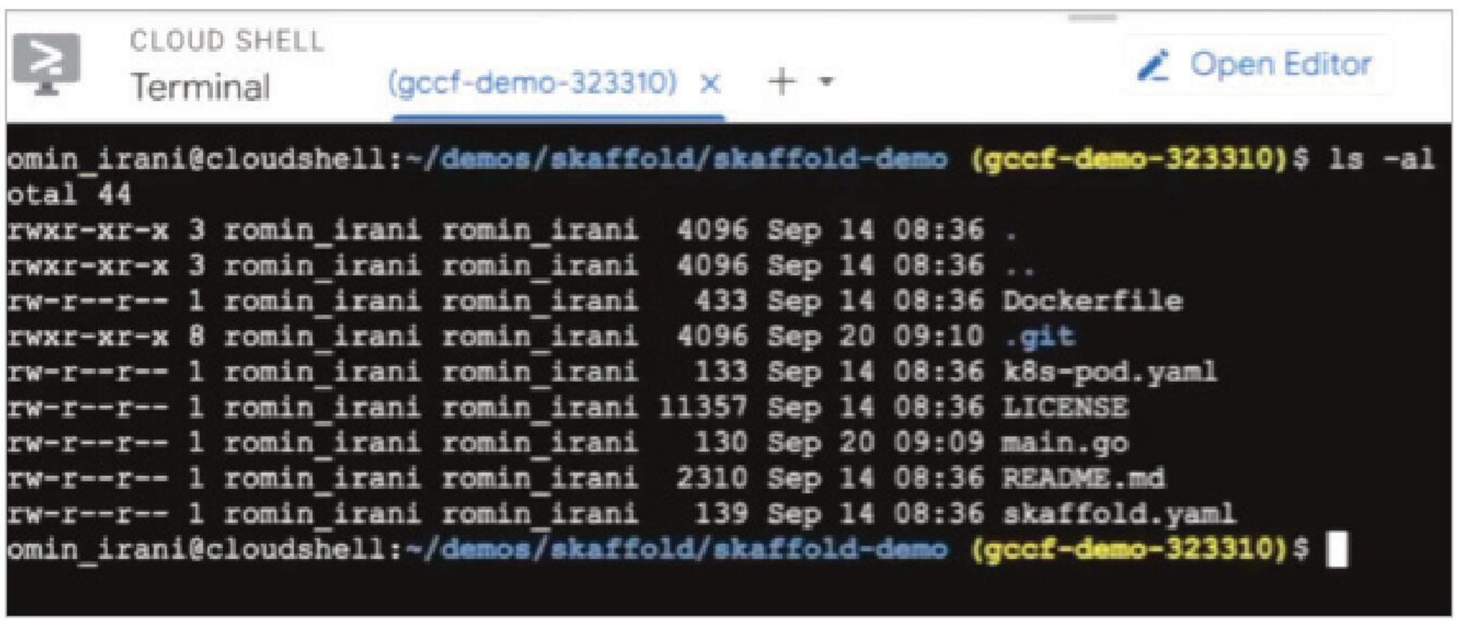


Figure 6: Inside the Cloud Shell terminal

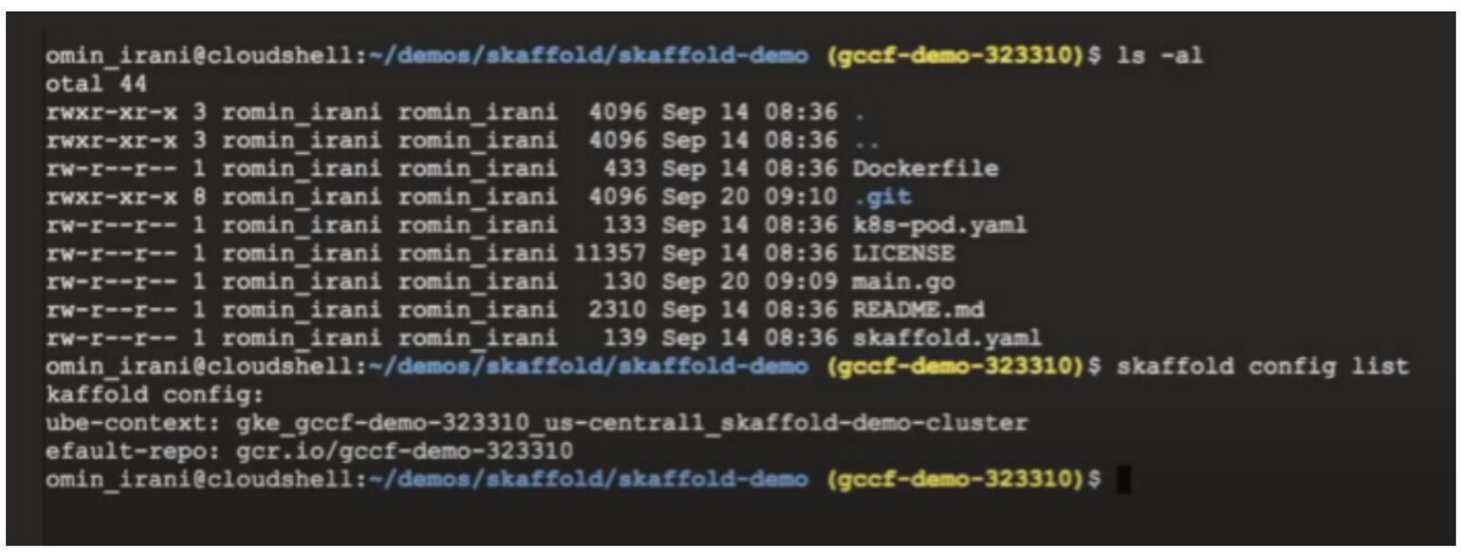


Figure 7: '$skaffold config list' command is applied in the terminal

a build perspective, as Skaffold really knows what it is supposed to build.

In the *pod.yaml* file (Figure 5), it is just stated that the image that is needed to be created is *Skaffold-example*. The *Skaffold.yaml* file will also show you the image that is needed to be built along with the deployment steps. All these steps will let Skaffold pretty much understand and detect the configuration.

In Figure 6, you can see that in the terminal inside the demo folder we have the *Skaffold.yaml* files, the *k8s-pod.yaml* and the *main.go*. Since the Skaffold tool is already installed as part of Google Cloud Shell, on typing *$skaffold config* list you can see (Figure 7) that the current cube context is set to that demo cluster, and the default-repo is also set to the Google container registry.

Now, we have a folder in which we have got our configuration file. All we need to do is run the developer end-to-end pipeline in Skaffold by using the *$skaffold dev* command. It will build everything up for the first time, and will begin with the deployment process. It will then provide you the relevant output and will wait for the code changes as well.

Skaffold is just like any other command-line tool that can be deployed on your machine or any other server that you desire; it comes preconfigured with Google Cloud Shell. All in all, Skaffold can automate a lot of steps that you would need to go through manually otherwise. Automating these steps will not only save time so that you can focus on other important things, but also make the application less prone to unwanted bugs and errors. END

By: Romin Irani

The author is developer advocate, Google Cloud.

Transcribed and curated by Laveesh Kocher.

Supervised Machine Learning with KNIME

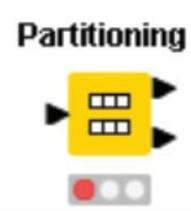
The KNIME (Konstanz Information Miner) analytics platform is perhaps the strongest and most comprehensive free platform for drag-and-drop analytics, machine learning, statistics, and ETL (extract, transform and load) models. This article will help develop a supervised machine learning decision tree algorithm through the KNIME tool for a car evaluation data set.



Data science, data pre-processing, modelling, analysis, and visualisation are all enabled within KNIME, the Konstanz Information Miner. The workflows can run both through the interactive interface and also in batch mode in this free and open source software. These two setups allow for easy local job management and regular process execution. KNIME integrates various components for machine learning and data mining through its modular data pipelining 'Lego of Analytics' concept.

Let's develop a supervised machine learning decision tree algorithm through the KNIME tool for the car evaluation data set. Decision trees are created for the car model based on the technical features, pricing criteria (overall cost, maintenance cost), comfort, size of luggage boot, safety, etc. The decision tree learner node and the predictor nodes are connected with both trained data and test data for training and evaluating the model with better output. Then the decision tree predictor is configured with the scorer node to generate the required evaluation performance metrics and to report data for user interface visualisation.

List of nodes

CSV Reader Node	
Color Manager Node	
Partitioning Node	
Decision Tree Learner Node	
PMML Writer Node	
Decision Tree Predictor Node	
Scorer Node	

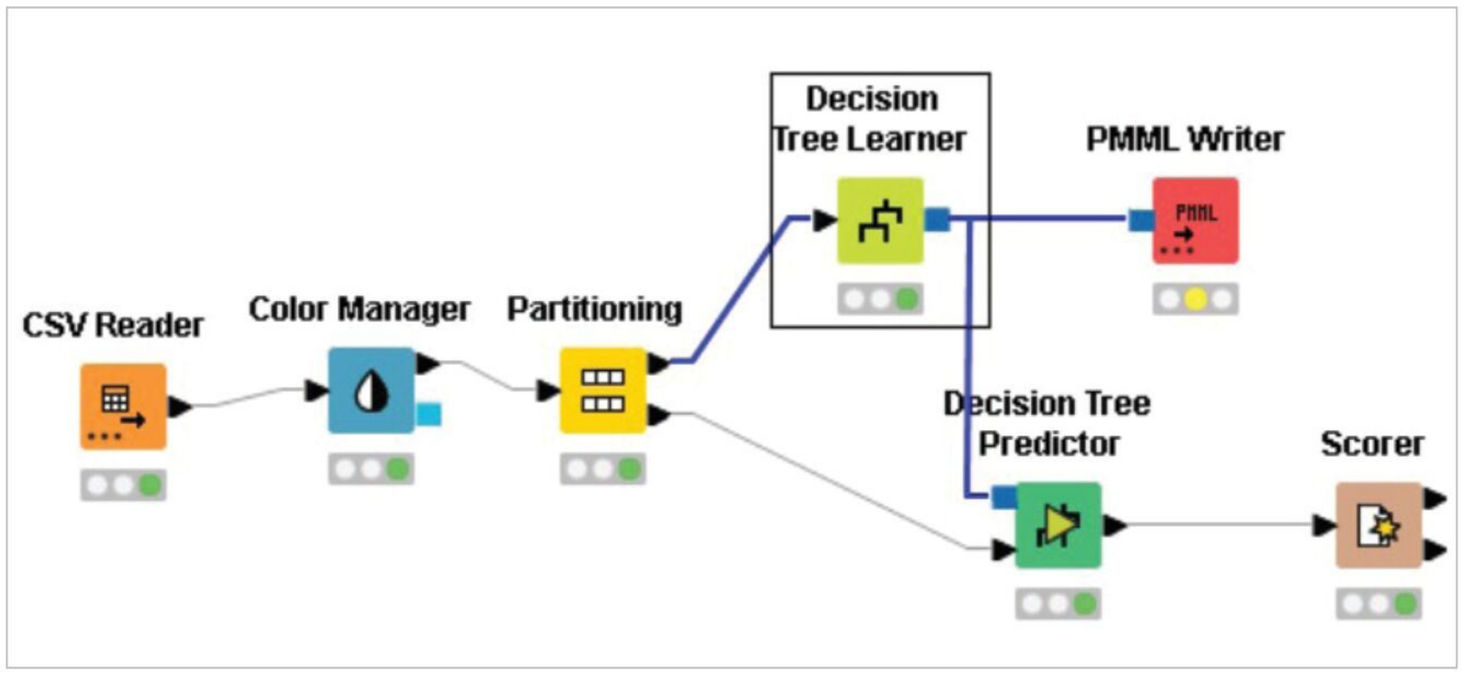


Figure 1: Car evaluation decision tree model KNIME workflow

To install KNIME:

- Download the KNIME Desktop Application from *https://www.knime.com/*.
- Install this application and provide the necessary permissions during the setup process.

Decision tree

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to construct a model that contains conditional control statements. Decision trees are used for handling non-linear data sets effectively. The decision tree tool is used in real life in many areas, such as engineering, civil planning, law, and business.

Designing a supervised ML car evaluation decision tree model

First, create a new workflow for the car evaluation model. Drag the *.csv* file into KNIME to create and initialise the *CSV Reader* node. Connect it to a colour manager, where each genre can be assigned a specific colour under 'swatches' or a preset under 'palettes'.

Next, connect the colour manager to the *partitioning* node. This is used to separate the given data set into two partitions -- one consisting of 80 per cent of the data and the other consisting of the remaining 20 per cent of the data. Thus the data is partitioned into two sets for testing and training in an 80:20 ratio. This partition is done by choosing the 'randomly' option.

Now we need to connect the *partitioning* node to the decision tree learner. Configure it as a class column as units, and change the records as per the required analysis. Next, connect the decision tree learner to the decision tree predictor. Send the smaller partition (test data) to the decision tree learner in order to classify the values, which can then be predicted by the decision tree predictor.

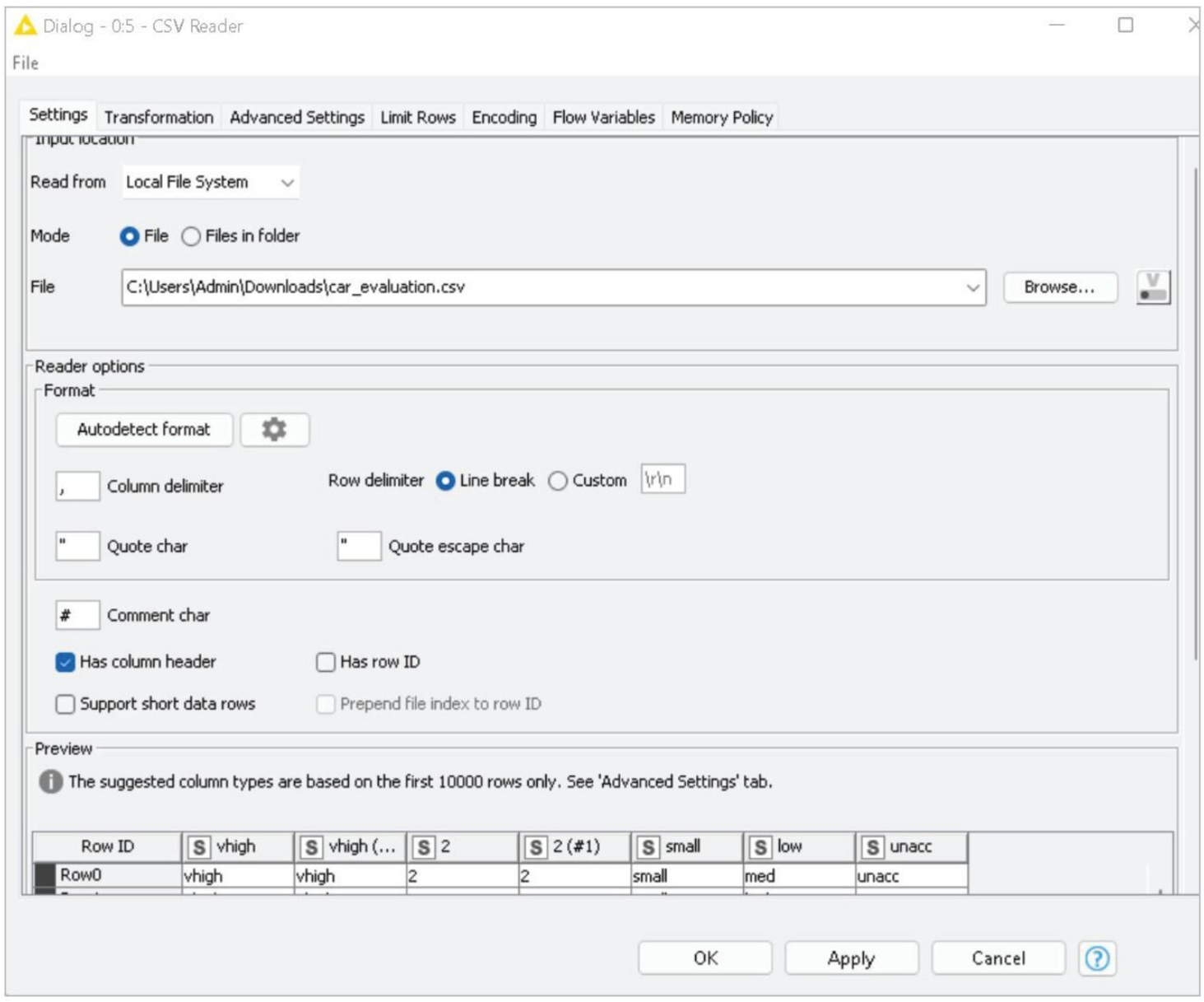


Figure 2: Configuration of *CSV reader* node

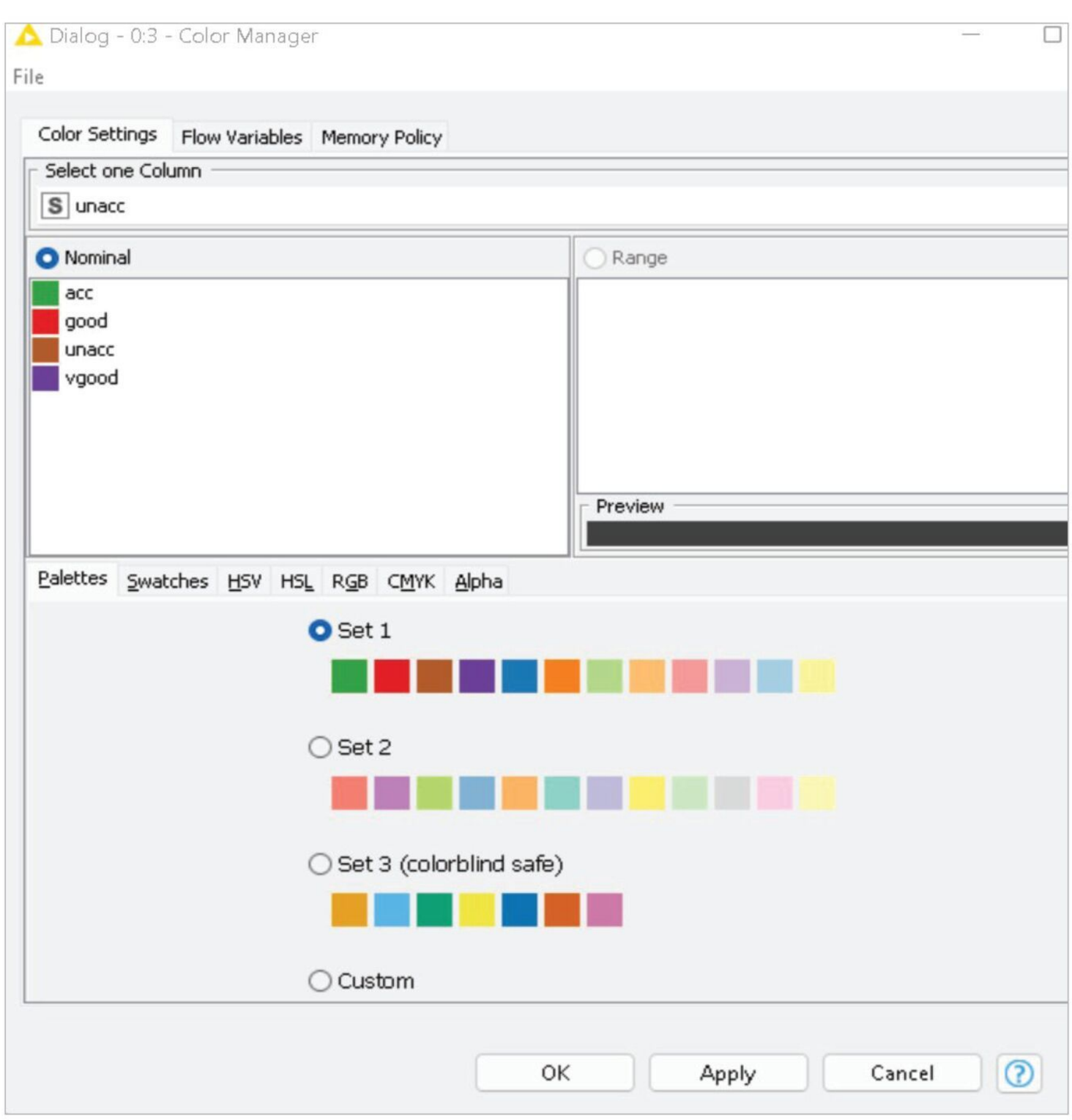


Figure 3: Configuration of *colour manager* node

Finally, send the output to the scorer node to measure the performance metric score. Compile and execute all the nodes at each stage of connection to get proper outputs.

Now, for viewing the output as a report, connect it to the PMML (predictive model markup language) writer node. Figure 1 shows the KNIME workflow.

Node configurations

The *CSV reader* node reads CSV files (Figure 2). Use this node if the workflow is created in a server or batch environment, and when the input files structure changes between different invocations. The car evaluation data set is loaded using the *CSV reader* node.

Colours can be assigned for either nominal (possible values have to be available) or numeric columns (with lower and upper bounds). If these bounds are not available, a '?' is provided as a minimum and maximum value. The values are then computed during execution. If a column attribute is selected, the colour can be changed with the colour chooser (Figure 3).

The input table is split into two partitions (i.e., row-wise), e.g., train and test data. In this case, 80 per cent of the data is used for training and 20 per cent for testing (Figure 4).

The *decision tree learner* node induces a classification decision tree in the main memory (Figure 5). The target attribute must be nominal. The other attributes used for decision-making can be either nominal or numerical. Numeric splits are always binary (two outcomes), dividing the domain into two partitions at a given split point. Here the target variable is 'Evaluation'.

The *decision tree predictor* node uses an existing decision tree (passed in through the model port) to predict the class value for new patterns (Figure 6).

The *scorer* node compares two columns by their attribute-value pairs and shows the confusion matrix, i.e.,

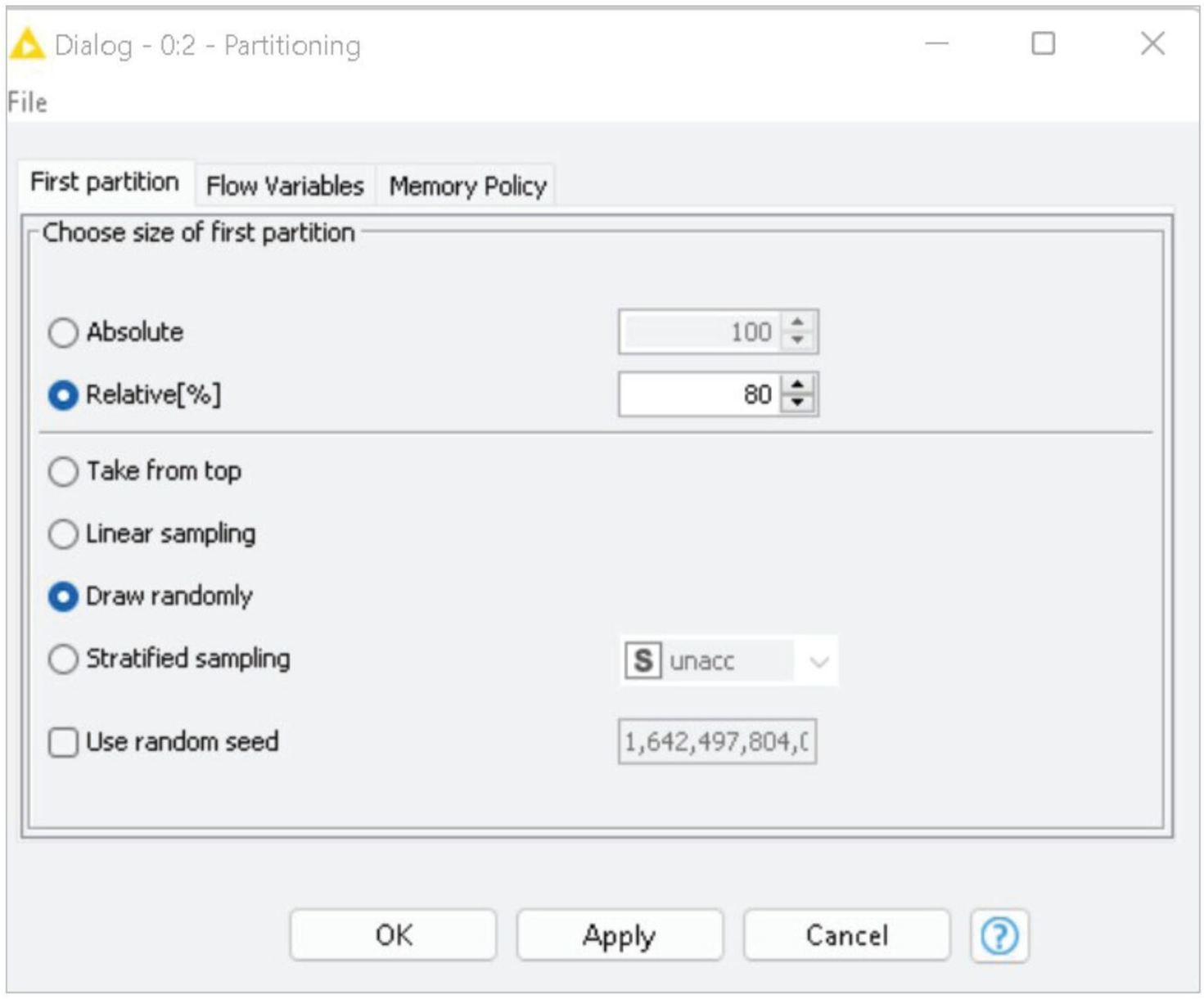


Figure 4: Configuration of *partitioning* node

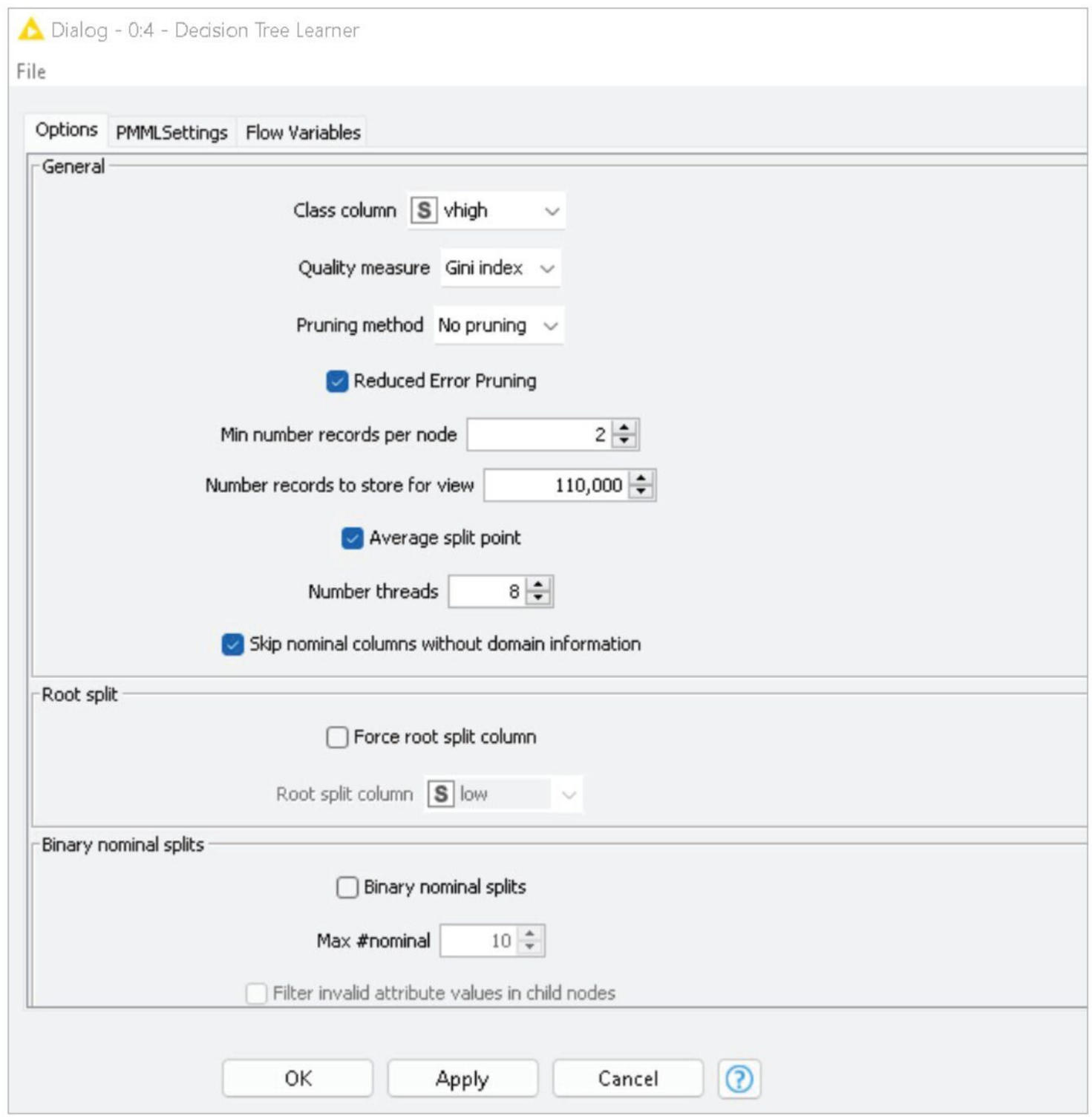


Figure 5: Configuration of *decision tree learner* node

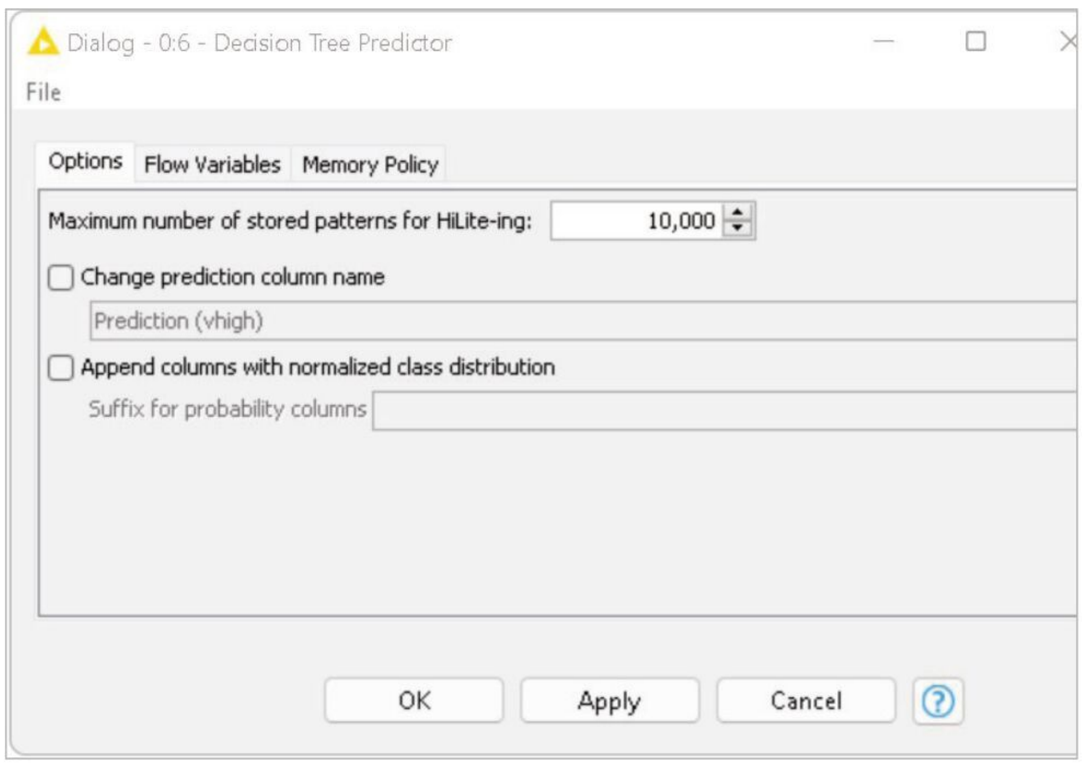


Figure 6: Configuration of *decision tree predictor* node

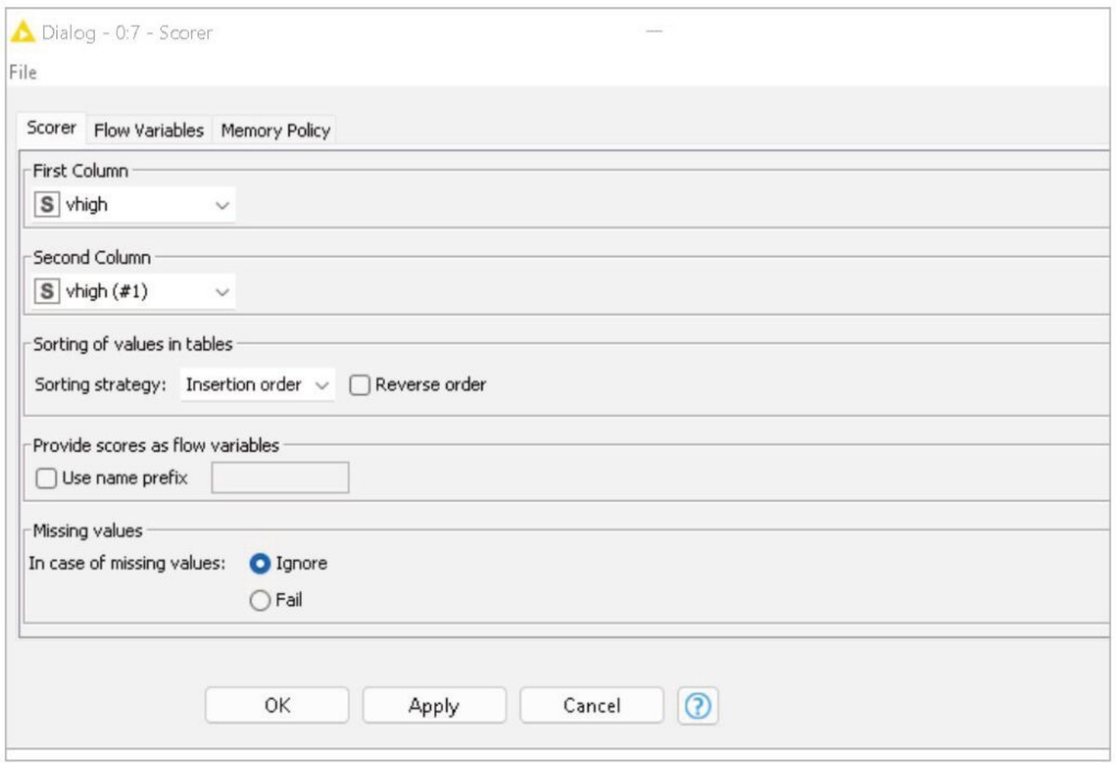


Figure 7: Configuration of *scorer* node

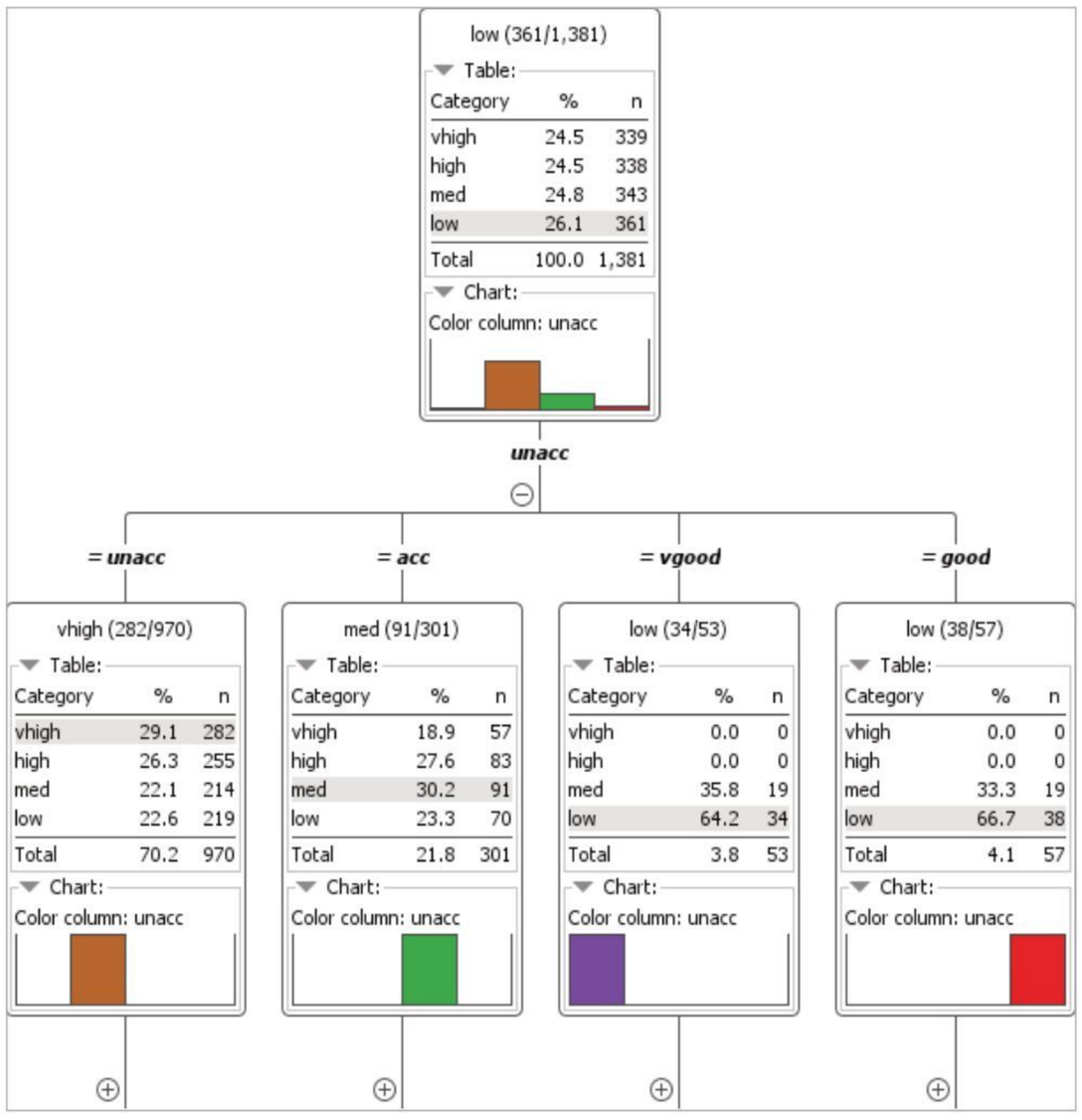


Figure 8: Decision tree output of the car evaluation model

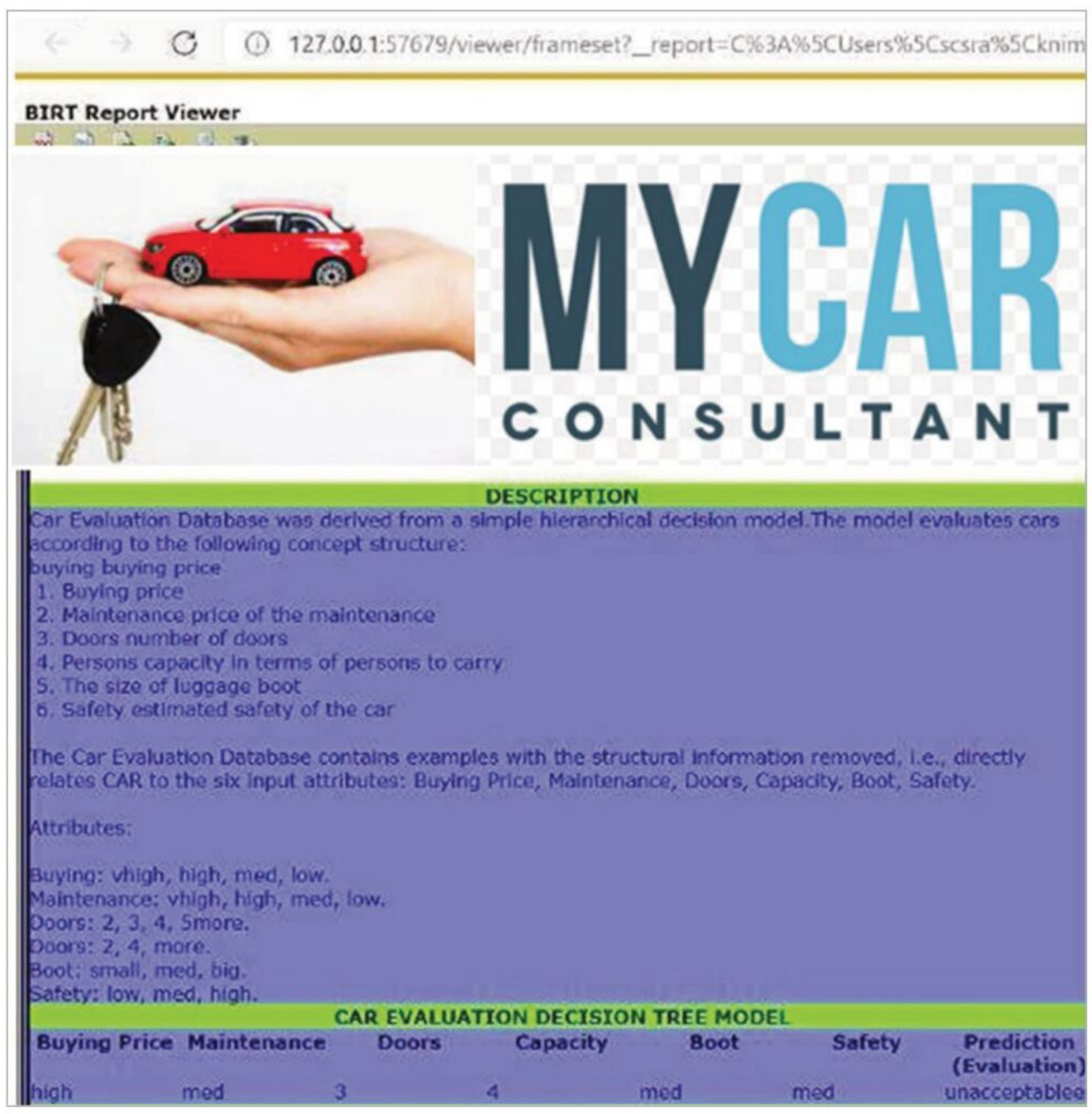


Figure 9: Visualisation report of car evaluation model

how many rows of which attribute and their classification match (Figure 7). Additionally, it is possible to highlight cells of this matrix to determine the underlying rows. Here, the accuracy achieved is 95.6 per cent.

Visualisation output

Figure 8 demonstrates the decision tree output of the car evaluation model, which visualises the learned decision tree. Based on the selections, the tree can be expanded and collapsed with the plus/minus signs.

As with the splitting criteria, the attribute name of the parent node is split, and its numeric value and nominal set value are represented to this child. The value in round brackets states *(x of y)*, where x is the quantity of the majority class and y is the total count of examples in this model.

Figure 9 represents the visualisation report of the car evaluation model. The PMML node is used to transmit models between nodes. PMML ensembles models, through reader and writer nodes, for importing and exporting these with other software. END

By: Dr Maheswari R.

The author is an associate professor in the Vellore Institute of Technology, Tamil Nadu. She has served as a resource person for hundreds of national and international workshops and conferences in the field of IoT.

Automatic Certificate Generation Using Python

Automation projects are nowadays highly appreciated due to the availability of cloud computing resources. This article is in continuation to the 'Automatic Certificate Generation Using MATLAB' DIY article published earlier and also available at *https://www.electronicsforu.com/electronics-projects/automatic-certificate-generation-matlab*, which garnered much attention because of the code's readability and scalability. Several academicians, especially students, contacted us regarding the project and its future scope. This motivated us to create a similar project using Python.

Python will help to overcome a few challenges in MATLAB. The project uses only online tools and packages, hence does not require any installation. Python language is used in all applications ranging from image processing to artificial intelligence. Figure 1 represents the basic idea about generating certificates using Python.

The Python script is run from an online tool called Colab, which accesses a sheet containing information, an image file that is a blank certificate and a ttf- font file to decide font type. When the script is run, a zipped folder is generated that can be downloaded to the local system from the browser. The folder contains all the certificates generated by the project. The code can be easily modified for generating any number of certificates and any number of fields to be printed on the certificates.

Advantages of Python on Colab

The project uses Colaboratory, Colab in short—a Google cloud based platform that provides a Jupyter notebook-like environment to write and execute Python code on the go. Colab offers some of the free services to the users, which are sufficient to carry out minor projects involving data science, artificial intelligence, machine learning, and deep learning. Apart from providing

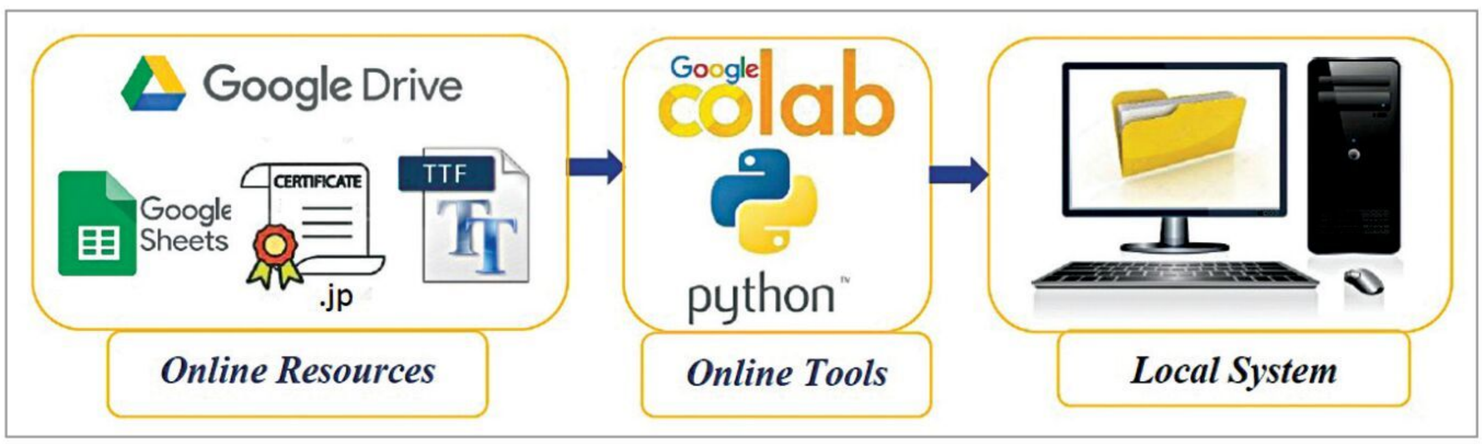


Figure 1: Block diagram of the project

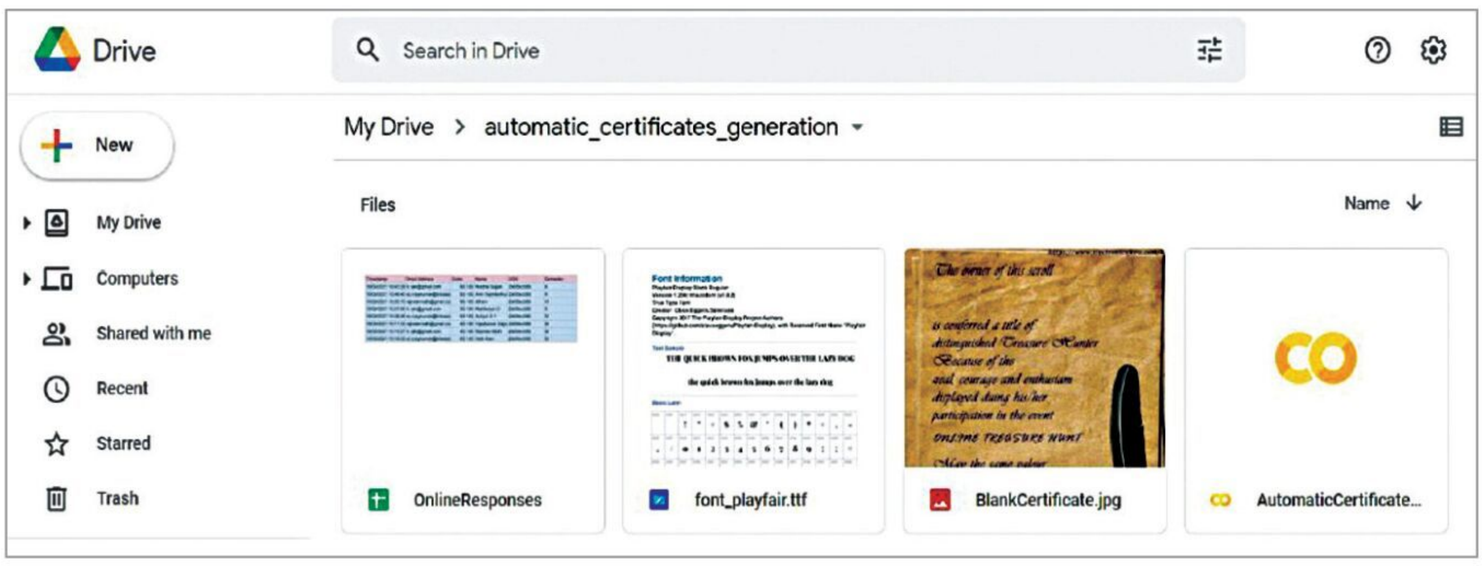


Figure 2: Work environment



Figure 3: Initiating Colab



Figure 4: Running the code in Colab

Requirements

Google Account (Gmail), Internet Browser, Internet and unzipping software (optional). Unzipping software is required to open and access the certificates which are in the zipped folder.

free runtime with around 12Gb RAM, it provides access to its GPUs and TPUs that can accelerate the training process when working with huge datasets.

Colab has almost all the standard libraries pre-installed, offering users the plug-and-play functionality that saves them from the headache of library dependency. Moreover, the files generated while executing the codes can be safely saved in the Google drive itself by just mounting it and giving permission to Colab to access your drive. These advantages of Colab makes it very popular among the beginners, researchers, and developers to build their projects and access them on the go.

Work environment

This section can be skipped by beginners in Python. Work environment is completely online and can be visualised by visiting the link below in any Internet browser such as

tinyurl.com/EFYdiyWE, or

https://drive.google.com/drive/folders/1qI7pOwR_vlm7TF9p5jZ-xU4--G4jEK9t?usp=sharing

The environment consists of four files as shown in Figure 2.

1. ***OnlineResponses:*** Google sheet containing participant information (provided by participants through a Google form created for an event).
2. ***font_playfair.ttf:*** True Type Font (.ttf) file which decides the font style being used for printing text on the image.
3. ***BlankCertificate.jpg:*** Image file consisting of the certificate.
4. ***AutomaticCertificateGeneration.ipynb:*** Ipython Notebook file where the code is to be executed.

Procedure to run the project

Step 1: Access Google Colaboratory by visiting the link below using an Internet browser like Google Chrome, Internet Explorer, or Mozilla Firefox. The longer version of URL (uniform resource locator) is provided in case tinyurl is deprecated such as

tinyurl.com/EFYdiy, or

https://colab.research.google.com/drive/1y_FMhXHHr0jnACirAc5VBTFowXL9iuYq?usp=sharing

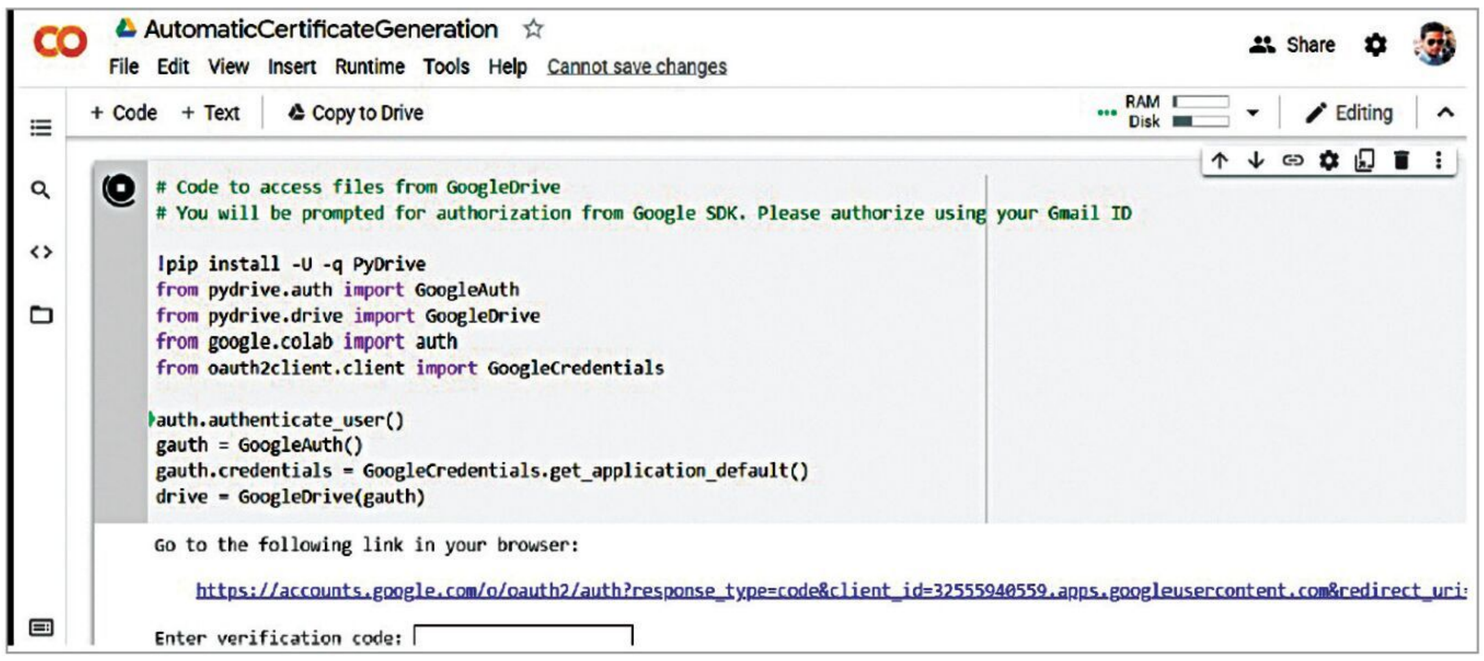


Figure 5: Authorising Colab to access drive

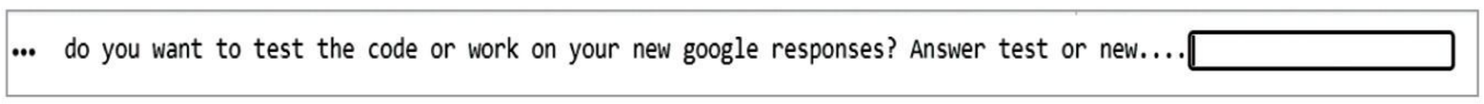


Figure 6: Testing the project



Figure 7: Successful generation of zipped file



Figure 8: Files generated and downloaded as a zipped folder

If not signed in, click on 'Sign in' button as shown in Figure 3 and join with your Google (Gmail) account.

Step 2: From the menu bar, select 'Runtime' and select 'Run All' as shown in Figure 4.

Step 3: Colab will ask for authorisation to access your Google drive using 'Google Cloud SDK' as shown in Figure 5. Click on the link generated in the code and complete authorisation by pressing Enter. This is a security step and has to be completed every time this project is executed.

Step 4: The code continues its execution and asks for one last input as in Figure 6. For testing this project, type 'test' (only four small-case letters) and press *Enter*. (After learning the project, customisation of the code can be done by typing 'new'. Customisation of the project is very simple and is discussed in a later section.)

Step 5: The code continues its execution and successfully generates zip file *Certificates_EFY.zip* as shown in Figure 7 and Figure 8. The certificates can be viewed by unzipping the folder using software like WinRAR, or WinZip.

Customisation of the project

For creating your own project, minor changes are to be made in the environment setup depicted in Figure 2. The Python code contains many headings and comments so that beginners in Python can also enjoy this project. Follow the steps given below.

Step 1: Make a copy of the code by using menu bar.

File→Save a Copy in Drive

Step 2: Open your Google drive by visiting the URL *drive.google.com*.

Step 3: Upload your Excel sheet or access Google sheet from Google drive. The sheet should contain information of participants. Click on *share* (View Only) and copy the shared link. Make any changes in the sheet as per your requirement.

Step 4: Run the copied Colab code and select 'new' when prompted. Paste the shared link of Excel/Google sheet and its file name in the fields as shown in Figure 9. In our example, the Excel sheet is named as 'Copy of OnlineResponses' and the shared link is: *https://docs.google.com/spreadsheets/d/1FSRBf8YFZSKDvK5_peSTxk2HCfJlXKkjDAdyjXPwTsQ/edit#gid=0. Here the id is 1FSRBf8YFZSKDvK5_peSTxk2HCfJlXKkjDAdyjXPwTsQ*

Step 5: For changing certificate image and fonts, change the code at various parts shown in Figure 10. Many free fonts and certificates are available online. You can modify the BlankCertificate.jpg image and

```
elif selectresponses=='new':
  print('file id of a Google Responses is part of URL and is obtained from URL as below')
  print('if URL is https://docs.google.com/spreadsheets/d/1WDmI18_XPp0JVW5CmBcimbgf3iY1IggfQ31siqWIp0E/edit#gid=0')
  print('https://docs.google.com/spreadsheets/d/<---------------file id---------------------------->/edit#gid=0')
  print('enter file id as 1WDmI18_XPp0JVW5CmBcimbgf3iY1IggfQ31siqWIp0E ')
  fileid=input('provide file id from URL=')
  csvid=input('provide name of responses file=')
  fileDownloaded = drive.CreateFile({'id': fileid})
  fileDownloaded.GetContentFile(csvid + '.csv', mimetype='text/csv')
  import pandas as pd
  df = pd.read_csv(csvid + '.csv')
  print('Google Responses saved in Pandas dataframe')
else:
  print('error: Enter test or new')
```

```
do you want to test the code or work on your new google responses? Answer test or new....new
file id of a Google Responses is part of URL and is obtained from URL as below
if URL is https://docs.google.com/spreadsheets/d/1WDmI18_XPp0JVW5CmBcimbgf3iY1IggfQ31siqWIp0E/edit#gid=0
https://docs.google.com/spreadsheets/d/<---------------file id---------------------------->/edit#gid=0
enter file id as 1WDmI18_XPp0JVW5CmBcimbgf3iY1IggfQ31siqWIp0E
provide file id from URL=1FSRBf8YFZSKDvK5_peSTxk2HCfJlXKkjDAdyjXPwTsQ
provide name of responses file=Copy of OnlineResponses
Google Responses saved in Pandas dataframe
```

Figure 9: First step in customising (Google sheet)

Load Blank Certificate and fonts from Google Drive

```
[10] fileDownloaded = drive.CreateFile({'id':'1UHtxwV3e_OyGQGopLz-PBVt9U1X3quOu'})
     fileDownloaded.GetContentFile('BlankCertificate.jpg')
     fileDownloaded = drive.CreateFile({'id':'1JfMjm5ZffEo2shtZko2LC793GB654iys'})
     fileDownloaded.GetContentFile('font_playfair.ttf')
```

Figure 10: Second step in customising (certificate and font)

Pixel positions can be obtained using any image editor. Ex: MS Paint

```
from PIL import Image, ImageFont, ImageDraw
df_name=df['Name']
df_usn=df['USN']
index = df.index
number_of_rows = len(index)
for i in range(number_of_rows):
  name_text = df_name[i]
  usn_text=df_usn[i]
  my_image = Image.open("BlankCertificate.jpg")
  image_editable = ImageDraw.Draw(my_image)
  name_font = ImageFont.truetype('font_playfair.ttf', 150)
  image_editable.text((250,250), name_text, (128,0,128), font=name_font)
  usn_font = ImageFont.truetype('font_playfair.ttf', 70)
  image_editable.text((250,500), usn_text, (128,0,128), font=usn_font)
```

Figure 11: Third step in customising (pixel positions)

text fonts. The 'id' parts can be obtained from the URL after sharing the font file and image file.

Step 6: Identify the pixel positions on the certificate where participants' information is to be printed. Change code at highlighted parts shown in Figure 11.

Step 7: Run the code to generate the customised certificates. You should be able to download the certificate in a zipped folder.

Results and future scope

This project is a complete online project and hence can be used with any configuration system, irrespective of operating system and hardware. The project is completely scalable and can be used to generate any number of certificates, any number of times. For intermediate to expert Python coders, the project offers a steep learning curve in the form of Python packages and their utilisation. Simple modifications to code can generate data analysis reports. Interested hobbyists can elaborate the code to create a website for this project using Python code deployment methods. END

By: Vijaykumar Sajjanar and Rajinderkumar M. Math

Both authors are assistant professors, Department of Electronics & Communication, BLDEA's CET, Vijayapur, Karnataka

The article was originally published in the December 2021 issue of Electronics For You.

Integrating Zeek with ELK Stack

Zeek is an open source network security monitoring tool. This article discusses how to integrate Zeek with ELK.



In the article titled 'Network Security Monitoring Made Easy with Zeek' published in the March 2022 edition of this magazine, we looked into the capabilities of Zeek and learned how to get started with it. We will now take our learning experience a bit further and see how to integrate it with ELK (also know as Elasticsearch, Kibana, Beats, and Logstash).

For this, we will use a tool called Filebeat, which monitors, collects and forwards the logs to Elasticsearch. We will configure Filebeat with Zeek, so that the data collected by the latter will be forwarded and centralised in our Kibana dashboard.

Installing Filebeat

Let's first set up Filebeat with Zeek. To install Filebeat using *apt*, give the following command:

```
sudo apt install filebeat
```

```
# ============================== Filebeat inputs ===============================

filebeat.inputs:

# Each - is an input. Most options can be set at the input level, so
# you can use different inputs for various configurations.
# Below are the input specific configurations.

# filestream is an input for collecting log messages from files.
- type: log

  # Change to true to enable this input configuration.
  enabled: true

  # Paths that should be crawled and fetched. Glob based paths.
  paths:
    - /var/log/*.log
    - /opt/zeek/logs/current/*.log
    #- c:\programdata\elasticsearch\logs\*

  # Exclude lines. A list of regular expressions to match. It drops the lines that are
  # matching any regular expression from the list.
  #exclude_lines: ['^DBG']
```

Figure 1: Filebeat config (a)

Next, we need to configure the *.yml* file, which is present in the *etc/filebeat/* folder:

```
sudo nano /etc/filebeat/filebeat.yml
```

We need to configure only two things here. In the *Filebeat Input* section, change the type to *log* and uncomment the *enabled: false* and change it to *true*. We also need to specify the path of where the logs are stored, i.e., we need to specify */opt/zeek/logs/current/*.log*

Once this is done, the first part of the settings should look similar to what's shown in Figure 1.

The second thing to be changed in the Elasticsearch output section is under *Outputs*. Uncomment the *output.elasticsearch* and hosts. Make sure the URL of the host and port number are similar to what you configured while installing ELK. We kept it as *localhost* with port number 9200.

In the same section, uncomment the user name and password at the bottom, and enter the user name and password of the elastic user that you generated while configuring ELK after installation. Once this is done, refer to Figure 2 and check the settings.

Configuring Zeek

Now that we have completed installing and configuring Filebeat, we need to configure Zeek so that it stores the logs in JSON format. For that, ensure your Zeek instance is stopped. If it's not, execute the command given below to stop it:

```
cd /opt/zeek/bin
./zeekctl stop
```

Now we need to add a small line in the *local.zeek*, which is present in the *opt/zeek/share/zeek/site/ directory.*

Open the file as root and add the following line:

```
@load policy/tuning/json-logs.zeek
```

```
# ---------------------------- Elasticsearch Output ----------------------------
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["localhost:9200"]

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
  username: "
  password:

# ------------------------------ Logstash Output -------------------------------
#output.logstash:
  # The Logstash hosts
 # hosts: ["localhost:5044"]
```

Figure 2: Filebeat config (b)

```
sudo nano /opt/zeek/share/zeek/site/local.zeek
  GNU nano 4.8                /opt/zeek/share/zeek/site/local.zeek
##! Local site policy. Customize as appropriate.
##!
##! This file will not be overwritten when upgrading or reinstalling!

# Installation-wide salt value that is used in some digest hashes, e.g., for
# the creation of file IDs. Please change this to a hard to guess value.
redef digest_salt = "Please change this value.";

# This script logs which scripts were loaded during each run.
@load misc/loaded-scripts
@load policy/tuning/json-logs.zeek

# Apply the default tuning scripts for common tuning settings.
@load tuning/defaults
```

Figure 3: *local.zeek* file

```
# Module: zeek
# Docs: https://www.elastic.co/guide/en/beats/filebeat/7.16/filebeat-module-zeek.html

- module: zeek
  capture_loss:
    enabled: true
    var.paths: [ "/opt/zeek/logs/current/capture_loss.log", "/opt/zeek/logs/*.capture_loss.json" ]
  connection:
    enabled: true
    var.paths: [ "/opt/zeek/logs/current/conn.log", "/opt/zeek/logs/*.conn.json" ]
  dce_rpc:
    enabled: false
  dhcp:
    enabled: true
    var.paths: [ "/opt/zeek/logs/current/dhcp.log", "/opt/zeek/logs/*.dhcp.json" ]
  dnp3:
    enabled: false
  dns:
    enabled: true
    var.paths: [ "/opt/zeek/logs/current/dns.log", "/opt/zeek/logs/*.dns.json" ]
  dpd:
    enabled: false
  files:
    enabled: false
  ftp:
    enabled: false
  http:
    enabled: true
    var.paths: [ "/opt/zeek/logs/current/http.log", "/opt/zeek/logs/*.https.json" ]
  intel:
    enabled: false
  irc:
    enabled: false
```

Figure 4: *zeek.yml* configuration

Refer to Figure 3 and make sure the settings are done correctly.

As we have changed a few configurations of Zeek, we need to re-deploy it, which can be done by executing the following command:

```
cd /opt/zeek/bin
./zeekctl deploy
```

Now we need to enable the Zeek module in Filebeat so that it forwards the logs from Zeek. Execute the following command:

```
sudo filebeat modules enable zeek
```

We are almost ready; in the last step, configure the *zeek.yml* file to mention what type of data is to be logged. This can be done by modifying the */etc/filebeat/modules.d/zeek.yml* file.

In this *.yml* file, we must mention the directory where these specified logs are stored. We know that the logs are stored in the current folder, which has several files like *dns.log*, *conn.log*, *dhcp.log*, and many more. We need to mention each path in each section. You can leave unwanted files by changing the enabled value to false, if and only if you don't want logs from that file/program.

For example, for *dns*, make sure the enabled value is *true* and the path is mentioned as:

```
var.paths: [ "/opt/zeek/logs/current/dns.log", "/opt/zeek/logs/*.dns.json" ]
```

Repeat this for the rest of the

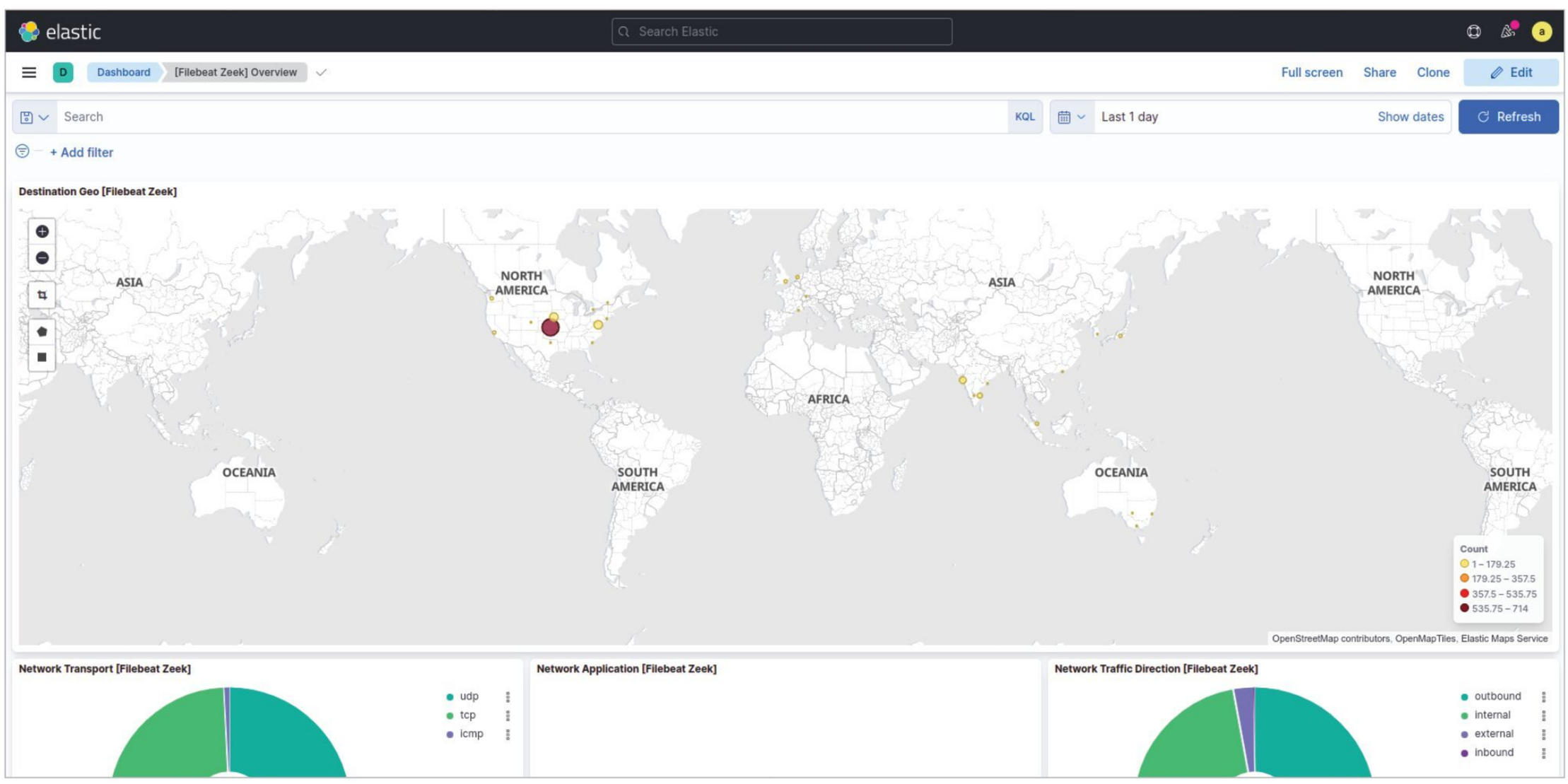


Figure 5: Dashboard of Kibana (Destination Geo)

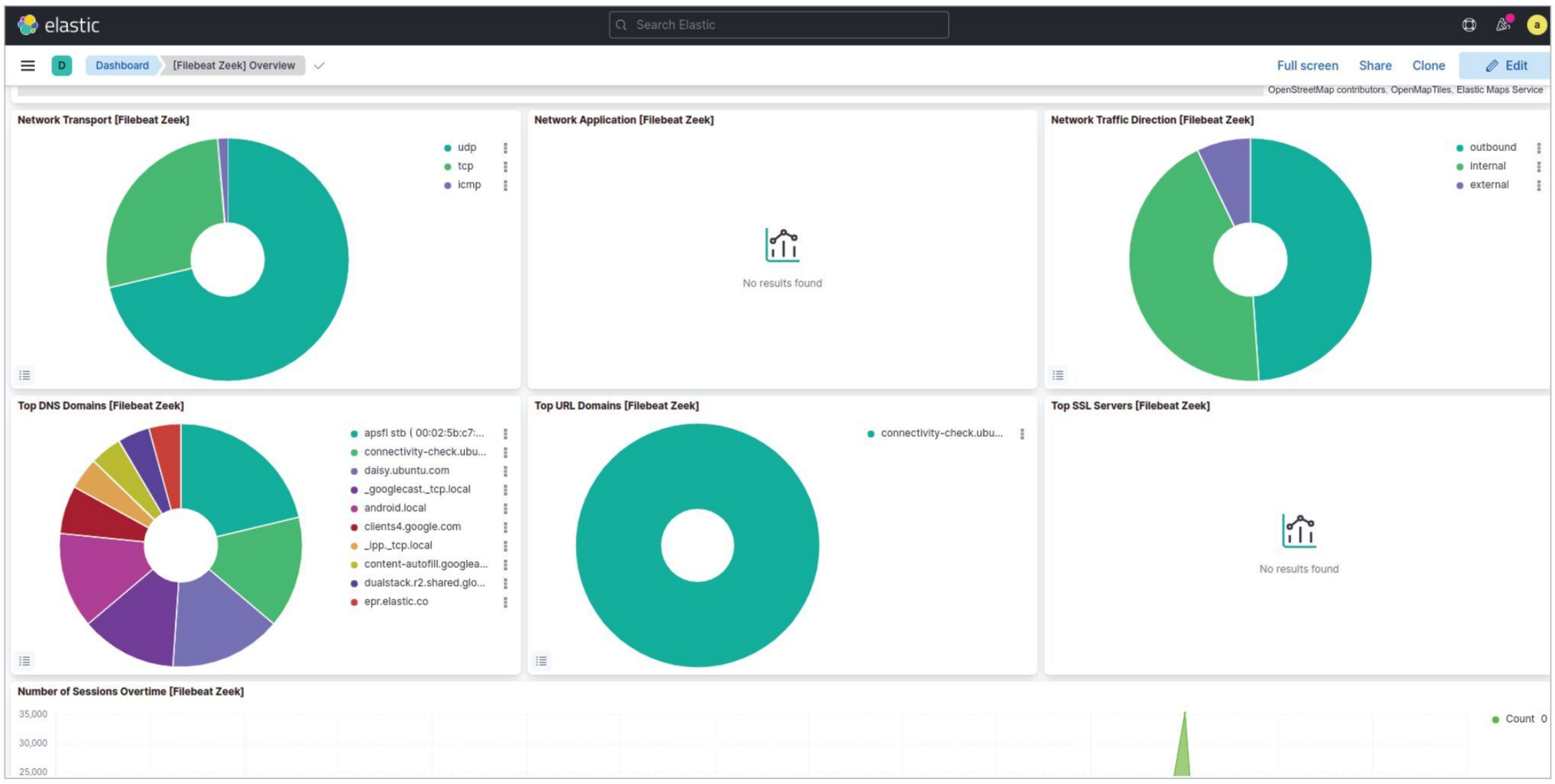


Figure 6: Dashboard of Kibana (Network)

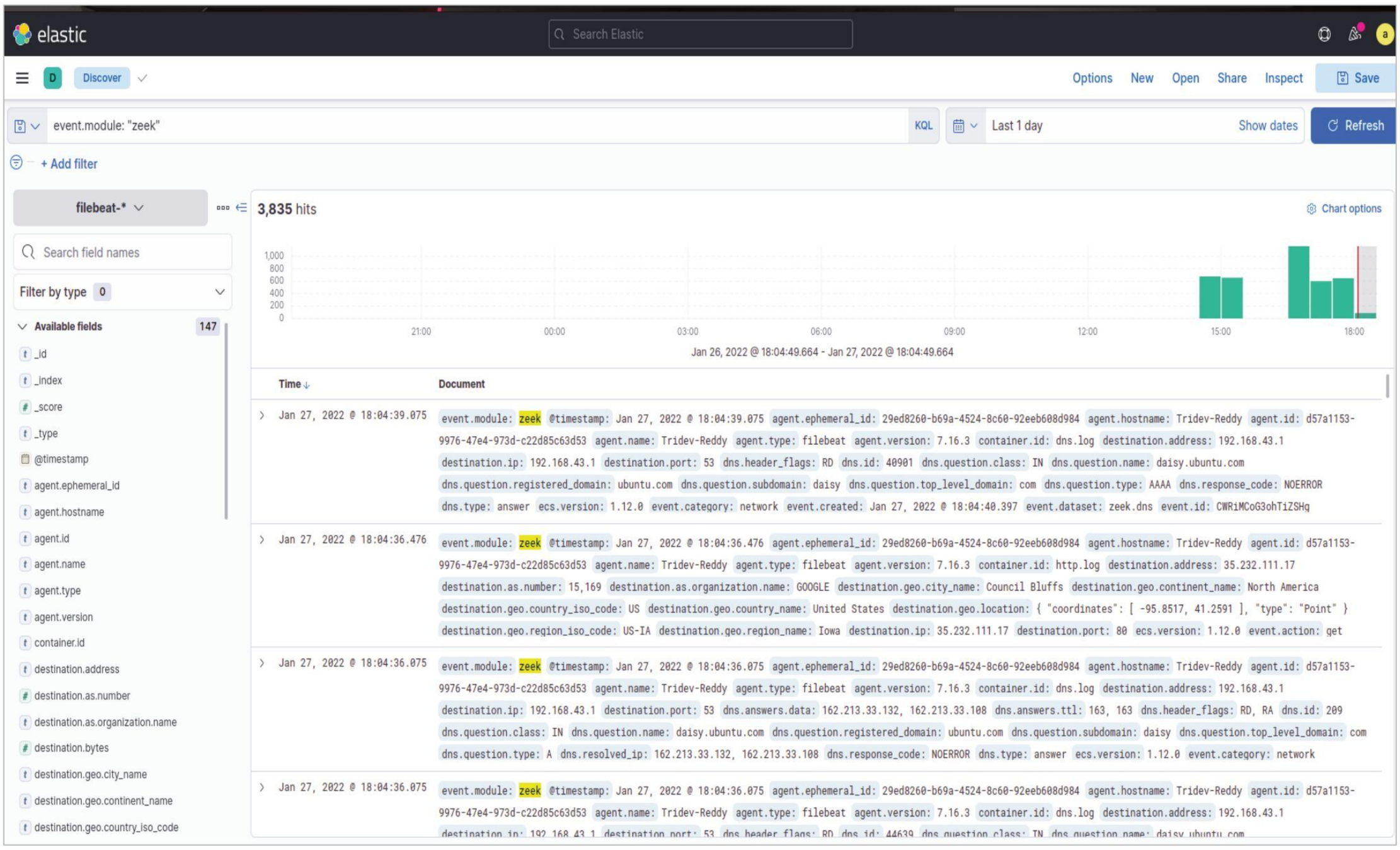


Figure 7: Filtered data by *event.module* query

files. We did this for a few that we needed. We added everything that was mainly required. You can do the same. Refer to Figure 4.

Now it's time to start the Filebeat. Execute the following commands:

```
sudo filebeat setup
sudo service filebeat start
```

Now that everything is done, let's move to our Kibana dashboard and check whether we are receiving the data from Zeek via Filebeat or not.

Navigate to the dashboard; you can see a clear statistical analysis of the data it has captured (Figure 5 and Figure 6).

Now let's move to the *Discover* tab and check the results by filtering using the query:

```
event.module: "zeek"
```

This query will filter all the data it received in a certain time and show us only the data from the module named Zeek (Figure 7).

Acknowledgements

The authors are grateful to Sibi Chakkaravarthy Sethuraman, Sudhakar Ilango, Nandha Kumar R. and Anupama Namburu at the School of Computer Science and Engineering, VIT-AP for their continuous guidance and support. A special thanks to the Center for Excellence in Artificial Intelligence and Robotics (AIR). END

By: Tridev Reddy and Aditya Mitra

Both the authors like to work in the field of network analysis, IoT and security.

THE COMPLETE MAGAZINE ON OPEN SOURCE
OpenSource ForYou
The latest from the Open Source world is here.
OpenSourceForU.com
Join the community at facebook.com/opensourceforu
Follow us on Twitter @OpenSourceForU

Should You Switch to Serverless Computing?

There's a lot of buzz around serverless computing. But should everyone shift to it? And why? Here are some insights.



Serverless doesn't refer to a non-existing server. Basically, serverless computing is a cloud-computing execution model where machine resources are allocated on demand by the cloud provider, which also takes care of servers on behalf of the customers. Hence, it can also be termed as a method of providing backend services as per usage. The advantage is that companies who get backend services from serverless vendors are charged based on usage and not a fixed amount of bandwidth or the number of servers used.

Why serverless?

In server based systems, it is important to manage infrastructure. This includes server/cluster provisioning, patching, operating system maintenance and capacity provisioning. Such management can be eliminated in serverless computing, helping us focus on just the application and its requirements.

Early physical systems had a lot of maintenance issues, added to the long time taken to deploy them. These were followed by servers being managed in virtual machines, but there was still a lot of complexity involved. Next came containerisation, which eased big and complex applications. But all these systems were server based. The idea of serverless has transformed computing infrastructure, as it does not need server management and can be scaled flexibly. Scaling can be done for customers who use the application, thus optimising costs. With flexible scaling, there is no idle capacity, which leads to high availability.

The rate of adoption of serverless computing is increasing every year, making public cloud providers and

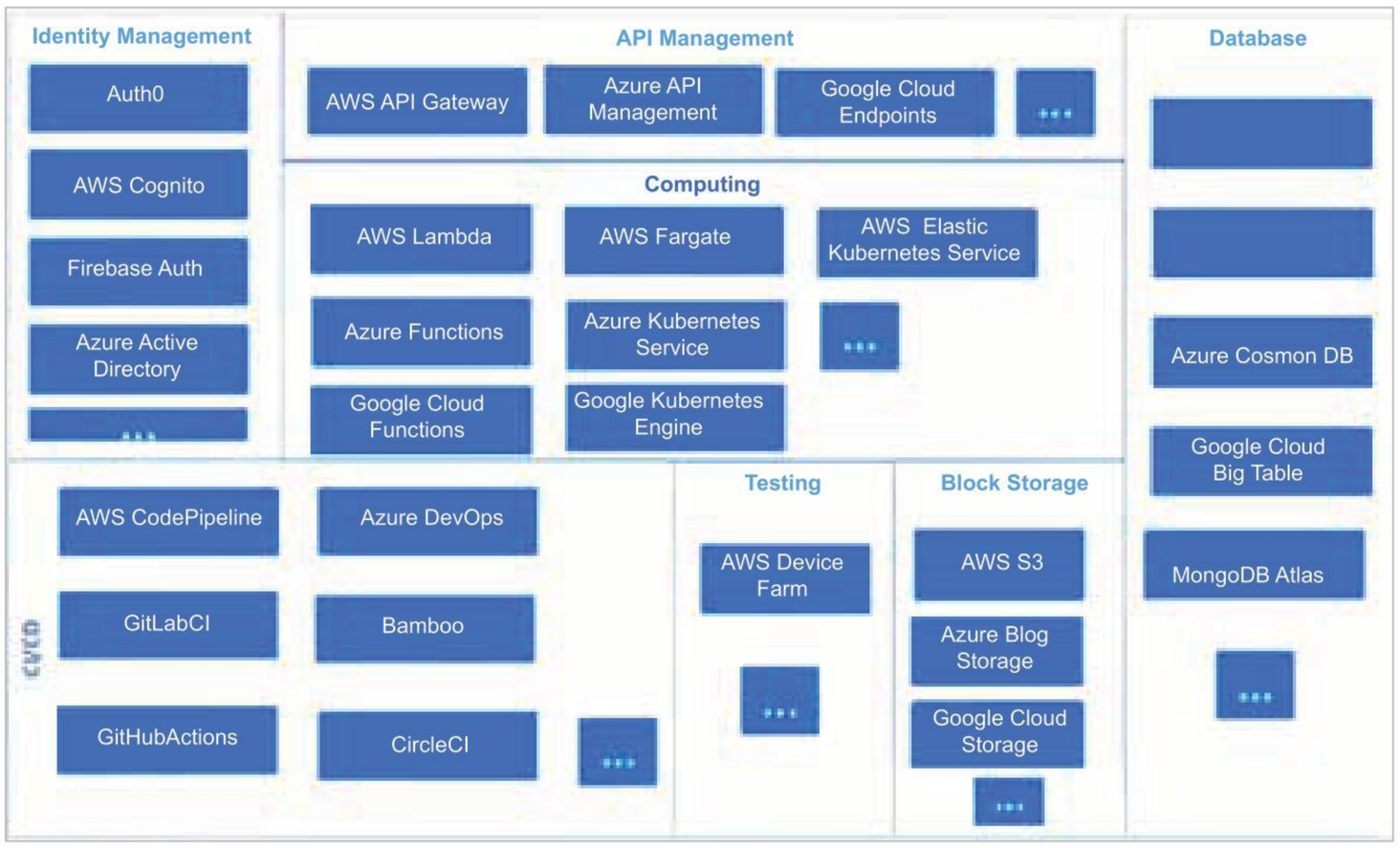


Figure 1: Serverless landscape

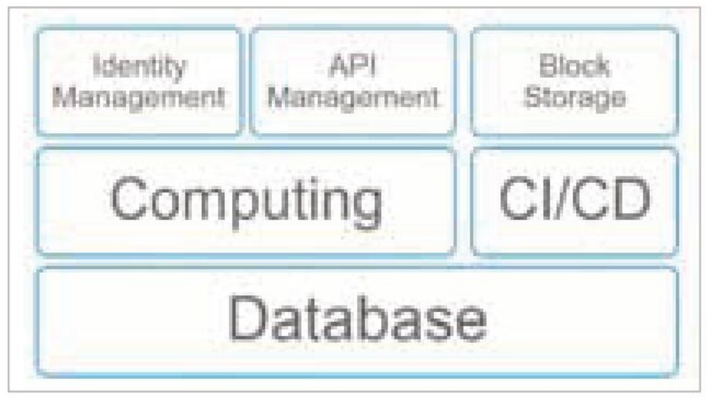


Figure 2: Serverless catalogue

The advantages of serverless computing:

1. Event driven computing
2. Cost efficiency
3. Smaller DevOps cycle
4. Lower management and compliance burden
5. Faster iterations and rollout of software and services
6. Enhanced reliability
7. Enhanced security; updates to security are rolled out in a timely way and do not affect the availability of services
8. Eliminates the need to time the scaling of capacity and resources; sudden peaks and drops in traffic are managed efficiently

other leading industries come up with newer serverless based mechanisms. Its offerings are not just limited to computing but also affect databases, storage, identity management, API management, continuous integration and deployment (CI/CD), and so on.

Many startups, where cost plays a significant factor, have considered serverless as their primary choice for cloud services.

Serverless computing is expanding in all directions and is not just limited to one area (Figure 1). This landscape is predicted to only increase and not decrease, as serverless computing reduces costs and is very efficient.

Serverless catalogue

Identity management: From a client perspective, identity management is crucial for any application and involves a lot of effort with respect to compliance. Serverless computing simplifies many things like saving development and maintenance time by integrating with infrastructure resources, both on the cloud as well as on-premise. Services available for this include AWS Cognito, Auth0, Microsoft Azure Active Directory, and Google Firebase Auth, which are especially powerful for sensitive data and help save a lot of time.

API management: With most enterprises migrating to an API-first approach for their solutions, API management is essential for building a distributed and scalable architecture. It provides a proxy for your computer storage layer. Many solutions also support microservices architecture. REST and GraphQL are popular for designing APIs. AWS API gateway, Azure API management, and Google Cloud Endpoints provide many default services. This is unlike yesteryears where developers had to take care of even minute services.

Computing: This involves two aspects, namely, serverless functions and containerisation. The functions expand and provide the ability to focus just on business logic and not on setting up the runtime environment. It works on event triggers like API invocation, message pushing in a queue, identity events, etc; integrates well with mobile as well as Web front-ends, and can scale up to millions of requests. Popular application runtimes like AWS Lambda, Azure Functions and Google Cloud Functions are supported by cloud providers.

Serverless containerisation involves containers that are fast becoming viable alternatives to virtual machines and reduce deployment issues. Cloud providers also provide serverless containerisation — for example, AWS Fargate, AWS Elastic Kubernetes Service, Azure Kubernetes Service and Google Kubernetes Engine. These services are quite powerful, and you only pay for what you use.

Database: Serverless computing offers the feasibility of databases, which are the fundamental building blocks of any application and help in storing structured and unstructured data. Both SQL and NoSQL databases are offered by cloud providers. They

Design principles to consider

- Stop guessing your capacity needs
- Test systems at production scale
- Automate to make architectural experimentation easier
- Allow for evolutionary architectures
- Drive architectures using data

also support different open source and proprietary database engines while drastically reducing operational, maintenance, scalability, and security efforts. Working on a database has been simplified through serverless computing. One good serverless database is AWS Aurora Serverless. It is cost-effective and supports both open source and proprietary software. AWS DynamoDB, Azure Cosmos DB, MongoDB and Google Cloud Big Table are also supported by serverless computing.

Block storage: Along with databases, block storages are also essential as they store media, and log files at scale. Large volumes of data, based on the type of application, can be stored by enterprises as well. With the need for storage space increasing, this serverless solution charges nominally as per usage, and is highly available and durable as well. AWS S3, Azure Blob Storage, Google Cloud Storage are some of the popular serverless block storages available.

Continuous integration and continuous deployment: Playing an important role in deployment of applications, a serverless CI/CD solution allows developers themselves to manage CI and CD (though automation is a one-time process). It saves time, is prone to fewer errors, and is readily available and cost-effective. The consistent server performance leads to a faster build process as businesses now expect quick releases. AWS Code, Pipeline, GitLab CI, GitHub Actions, Azure DevOps, CircleCI and Bamboo are a few serverless CI/CDs.

Serverless is a paradigm shift

"Serverless is not for all use cases, but for only a few that can be implemented. It can be only be a part of a solution, not the entire solution." This is a myth and not a fact.

Serverless computing can be implemented in mobile applications by placing a simple API gateway. Real-time analytics can also be done by utilising a combination of different services.

Serverless is popular and powerful when it comes to usage. Triggering event driven platforms like SNS, S3, DynamoDB, and Lambda can result in customised performance and execution. The Lambda layer is very powerful for implementing some applications.

Due to its simplicity, serverless computing needs a few design principles (see box). It must be designed smartly by architect developers since even though each service is different, assembling all of them in the right fashion is what matters. Earlier, people would guess a capacity and design the service accordingly. However, if something did change from the capacity perspective, a huge rework had to be done in setting up the infrastructure. Instead of being given an architecture system and expecting it to survive for the next five years, evolution of services makes experimentation easy in serverless computing. So it has given engineers and developers new parameters to think and plan accordingly. Hence, serverless computing is a paradigm shift and not just a technology.

Myths about serverless computing

1. *Serverless comes with latency*: Giving cost-effective solutions along with a provision to implement these services doesn't mean there is always a latency. There are ways to fix this, which unfortunately many are not aware of.
2. *Only compute is serverless*: Based on the landscape and catalogue of serverless, it is quite obvious that this is not the case. Many of us use serverless in various aspects but are not aware of it.
3. *All serverless APIs are microservices*: All microservices are not serverless though it can be used for the same.
4. *Serverless is only for limited use cases*: As per the different architectural flavours seen, we can conclude that this is not true.

Amidst all that has been spoken about here, the privacy of data being processed is not compromised in serverless, as all cloud providers give a provision to implement a certain way of keeping data private. AWS Cognito is both popular and powerful for data privacy. Similarly, for databases and transition of data, there are provisions to keep the written data in an encrypted way. In conclusion, serverless is a shift in mindset and not just a technology that developers and architects need to understand. END

By: Raghvendra Singh Dikhit

The author is senior solutions architect at Impetus Technologies. He is a technical thought leader, who is passionate about technology and technical solutions. He has experience in the IT field, mainly in the design and development of enterprise solutions. This article is based on a talk he gave at OSI 2021, a virtual event.

Transcribed and curated by Sharon Abhignya Katta

WiFi 6 Promises Much More than Faster Speeds

WiFi 6 increases the network connectivity, and has been developed to ensure the trillions of devices connected in the near future continue to work seamlessly and efficiently. Though it was certified in 2019, it faced a few challenges in testing due to the pandemic. This article gives an overview of this technology.



The next-generation standard in WiFi technology, termed ‘WiFi 6’, is also known as ‘AX WiFi’ or ‘802.11ax’. Developed to meet the exponential increase in demand for devices in the world, it can be used for virtual reality (VR) and smart home devices as well. It is an improvement on the current 802.11ac WiFi standard and meets current challenges in terms of capacity, efficiency, coverage and performance.

Researched in 2014, this standard was invented in 2018 and launched by the IEEE High Efficiency WLAN Study Group (HEW SG). It began certifications in late 2019, with Samsung’s Galaxy Note 10 and Ruckus R750 employing this technology. Specified to operate between the 1GHz and 6GHz bands, WiFi 6 focuses mainly between the 2.4GHz and 5GHz frequencies.

In an average household today, nine devices are connected to WiFi. WiFi 6 aims at improving the network rather than boosting the speed for individual devices.

Features of WiFi 6

***Multi-user, multi-input, multi-output (MU-MIMO)*:** This communication between routers and multiple devices concurrently. It supports four simultaneous data streams, added to which one user can have a considerable bandwidth of incoming data from a smart router, both on the 2.4GHz and 5GHz frequencies.



Figure 1: WiFi 6

1024-QAM: This helps WiFi 6 encode more bits per packet. There is a 25 per cent increase in throughput. Not only does it improve efficiency in high-traffic situations, it also maximises data rates. This is a huge advantage for modern enterprise applications.

Orthogonal frequency-division multiplexing (OFDM): This allows four times as many subcarriers and increases speed by 11 per cent. The expanded signal allows for greater simultaneous packet delivery across users. Hence, the wait time between packets and latency is reduced.

Increased channel width: The 160MHz channel communication is added to the 80MHz band, thus doubling the channel width. This allows routers to handle more users and provide larger streams per user.

Base service station (BSS) colour: This system identifies adjacent networks and marks them. Signal filtering is made easy and network conflicts are minimised. In enterprise systems, it allows segregated networks to cover the same physical area without losses due to interference.

Target wake time (TWT): This feature is unique to WiFi 6. It allows each device to independently negotiate wake time for transmission and reception. This helps to increase total sleep time and maximise battery life. TWT enables many additional networking options, especially for IoT devices.

***Improved security*:** All WiFi 6 devices will need to include Wi-Fi Protected Access 3 (WPA3). This will lead to encryption of unauthenticated traffic, robust password protection against brute-force dictionary attacks, and superior data reliability for sensitive information with 192-bit encryption.

Beamforming: With eight support antennas, beamforming helps to improve data rates, and the range is extended by directing signals towards specific clients at once. It offers a backup for rapidly moving devices that may face issues with MU-MIMO. Beamforming also helps to control transmissions from antennas that cause signals to interfere on purpose. The signal can then be redirected to a new direction.

Devices that support WiFi 6

Until recently, WiFi 5 was the standard used for routers, repeaters, mesh networks and many WiFi clients. WiFi 6 was launched in 2019. There will be some compatibility issues for the earlier devices that supported WiFi 5 — they will be able to utilise the WiFi 6 network but not be able to receive support for the same.

WiFi 6 routers are backward-compatible, and it is better to make sure that the network is ready for that.

WiFi 6 enables lower battery consumption, making it a great choice for any environment, including the Internet of Things (IoT). It reduces unnecessary data activity, and tells devices when to put their data to sleep and when to be active. As a result, unnecessary data activity is reduced, and performance and battery life are maximised.

The Samsung Galaxy Note 10 and Ruckus R750 were the world's first smartphone and access point certified to support Wi-Fi 6, with the latest generation of the Apple iPhone following suit. The Wi-Fi Alliance has set up its certification programme, and new wireless products hitting the market are expected to start applying for compliance certification. The devices listed below are already WiFi 6 enabled:

- iPhone 11 and after
- Samsung Galaxy S10, S20, Note 10, and Note 20
- Apple computers with M1 processors
- Smart TVs

To take advantage of the improvements in the 802.11ax standard fully, both hardware and software functionalities have to be built on this WiFi technology.

Hardware testing

To unlock the full potential of the latest devices, a WiFi 6 router is needed to run the network. This was an expensive affair a few years ago, but now we have a number of options even for mesh systems, gaming routers, range extenders, and more. The best purchase can be made only when hands-on testing is done. Beating all its competitors, the current king in terms of speed for WiFi 6 routers is TP-Link Archer AX6000. This router was able to transmit data wirelessly at a rate of 1523 Mbps up to a distance of 1.5 metres (5 feet).

One important thing to remember here is that these routers do not magically increase speeds. The theoretical maximum of achieving 9.6 Gbps is unlikely. This high theoretical speed can be split up across a whole network of devices.

WiFi 6 emphasises quality connectivity in areas where connected devices are densely populated. It does not increase the speed of each device exponentially but ensures these operate at an optimum level.

Only the combination of a faster plan from the Internet service providers (ISPs) along with the WiFi 6 router, can fulfil its true potential. The real challenge is for the ISPs, as they need new fibre rollouts to capitalise on this next-gen technology. An important question is: when faster ISP speeds come, will the existing hardware become redundant?

Applications of WiFi 6

Large public venues (LPVs): Stadiums and convention centres are a few of the common areas where thousands of devices connect to WiFi at the same time. WiFi 6 can help to improve attendee experiences, increase customer interactions, and create value-added services like viewing instant replays or ordering food from one's seat at an event. WiFi 6 allows LPV owners to create new business opportunities.

Transport hubs: Public transport stations are also an area where people attempt to connect to the network simultaneously. OFDMA and BSS colouring in WiFi 6 provide the necessary tools needed to overcome this challenge.

IoT and smart city deployments: Power efficiencies in WiFi 6 enable IoT devices to go into sleep mode and turn on their transmitters at predefined intervals to prolong field time without much maintenance.

Education: Libraries, auditoriums, and lecture halls at college and university campuses have the highest density of WiFi users during the day, and almost no one at night. WiFi 6 is a perfect choice in this situation.

Top WiFi 6 solutions providers for enterprises

WiFi 6 is targeted for areas that have many devices connecting to one network, and is more than just a collection of access points. It enables superior security and performance, centralised configuration and management, and a much higher capacity for user density.

Arista's cognitive WiFi 6 platforms remove any single points of failure for seamless and limitless scaling. Its CloudVision WiFi uses Big Data analysis to proactively identify and remediate inefficiencies that end-user wireless experiences often encounter.

Cisco Meraki uses WiFi 6 multigigabit technology, allowing you to optimise wireless access with faster connections, greater user capacity, and more coverage. It also includes integration benefits for organisations that have other Cisco and Meraki components on their network.

Ruckus, which is currently owned by parent company Arris, not only sells WiFi 6 hardware products but also provides software solutions. Its 'ultra-high density technology suite' is designed to improve network performance in high density areas.

Extreme Networks, now acquired by Aerohive, specialises in on-premises wireless controllers and WiFi 6 compatible distributed architecture. This provides administrators the authority to tune and control networks with added resiliency when compared to cloud managed alternatives.

Fortinet is known for its security-centric systems. Admins can manage via a dedicated, on-premises WiFi 6 compatible controller or by using a cloud-managed service option. FortiNAC is a network access control platform that integrates into any wireless architecture options to provide access and monitoring of wireless users, as well as autonomous IoT devices.

Top Wi-Fi 6 solutions providers for homes

Deploying a Wi-Fi 6 router can be challenging, as the pricing is based on hardware specs, performance capabilities, and features. Budget Wi-Fi 6 routers — typically, bare-bones and dual-band models — range in price from US$ 70 to US$ 200. They use low-end components such as dual-core CPUs that offer low data rates, and lack features such as USB ports, anti-malware software, and multi-gig LAN ports. A high-end Wi-Fi 6 router costs more than US$ 600.

WiFi 6 whole-home mesh networking system routers come at a very economical price, and are being chosen by vendors who provide home-based solutions. It is easier and quicker to operate them when compared to adding a third-party wireless range extender to the network.

TP-Link Archer AX6000 Wi-Fi 6 router delivers speeds up to 5952 Mbps and comes with eight external antennas that provide a stronger connectivity throughout your house. Its features include band steering and airtime fairness, which direct clients to the less congested band and optimise time usage, respectively. It comes with a security system built-in for parental controls as well.

Asus AX3000 Wi-Fi 6 router has a 160MHz bandwidth and a total networking speed of around 3000 Mbps. It helps to improve the battery life of the connected devices with its power saving feature. Since it is compatible with the Asus AiMesh, it offers seamless connectivity to all corners of your house.

Netgear Nighthawk Pro Gaming 6-Stream Wi-Fi router is basically designed for gaming. Its design handles VR gaming, 4K streaming and more. With a WPA3 encryption, traffic controller firewall (apart from the anti-virus), anti-malware and data protection, it offers excellent security.

The challenges

WiFi 6 does not promise an increase in speed, but is an upgrade designed to make sure the speeds of our devices within a given range/area doesn't slow down a few years down the road. There are three major challenges it faces though, which are often overlooked.

Improving the functionality of unsupported devices: Even though WiFi 6 is backward- compatible, justice to it can only be done when this technology is used to the maximum. This means devices need to be upgraded each time.

Speed and performance outside the internal network: WiFi 6 can provide excellent connectivity for services like cloud file shares. However, the assets and resources of ISPs can affect speed and performance.

Coverage issues: Transmission and bandwidths are capped according to the regulations prevalent in each country. Hence, the coverage of WiFi 6 may be restricted to ensure this cap is met.

In spite of these challenges, companies like Aruba, Asus, AT&T, Boingo, Broadcom, Cisco, Comcast, CommScope, Cypress, Extreme Networks, Intel, Netgear, Orange, Qualcomm, TP-Link and Xiaomi are all focusing on the potential WiFi 6 has. END

By: Sharon Abhignya Katta

The author is a member of the editorial team at *OSFY*. She is passionate about innovation, especially in the field of telecommunications.

SSL Certificates: Make the Right Choice

Increasingly sophisticated techniques are being used to hack into data today. So it has become extremely important to use even better ways to protect your customers' data. SSL certification is one such way. This article looks at the different types of SSL certificates and how to choose one that suits your organisation.

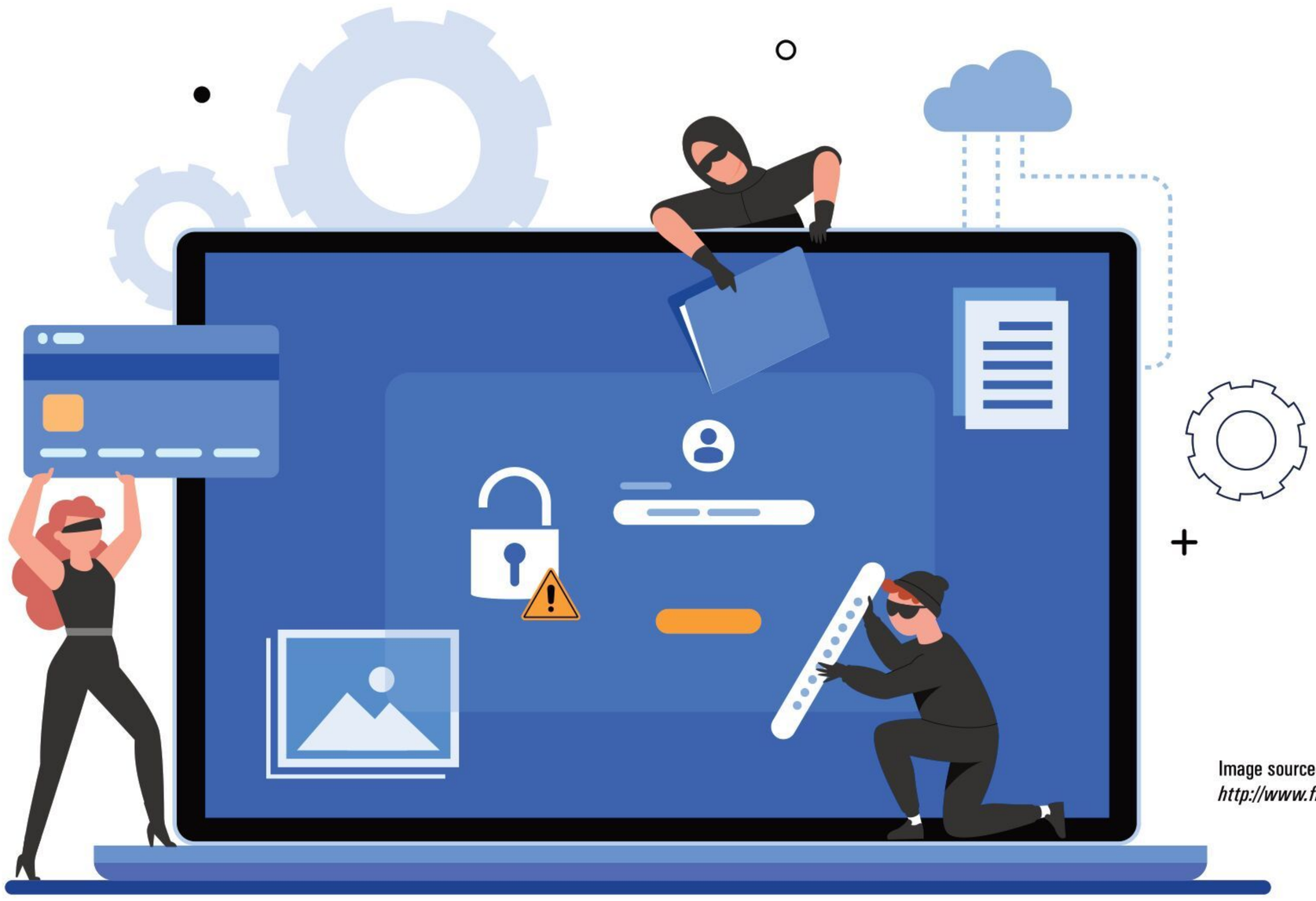


Image source: *http://www.freepik.com*

SSL certificates are an apt solution for securing data in transit. They create an automated mechanism to encrypt the transfer of data between the server and browser (or site and visitor). This encryption can only be decrypted by the intended system. So it discourages hackers from stealing data, and offers complete security against transit attacks on it.

Along with securing communication, SSL certificates also build trust with customers and help you emerge as a credible business by displaying virtual trust icons like a padlock, HTTPS prefix, and a green address bar. So let's learn about the major types of SSL certificates and how to choose the one that works best for you.

SSL certificate options: Similarities and differences

Let's first discuss the basic similarity all SSL certificates share. Regardless of their type, price and scope, every SSL certificate encrypts the to-and-fro communication between the site's server and the Internet browser of your visitor.

The difference lies in the terms and conditions of the validation process and the number of domains/subdomains it covers, etc. Proceeding to purchase an SSL certificate without first reviewing your present and future needs may result in costly mistakes, and you may end up paying a hefty amount for advanced features that are irrelevant to your business. Or you may get tempted to buy a cheap certificate that isn't a perfect fit for your business needs.

Due to the crucial role it plays, an SSL certificate is issued by an official certificate authority after a thorough verification procedure to confirm the legitimacy of ownership. So, this certificate not only encrypts the communication between websites and audiences, but also ensures the legal position of a site. It plays a vital role in building trust.

As you have now understood why we need a SSL certificate, we can discuss the different types of popular SSL certificates, along with their benefits and issuance procedures.

Domain validation (DV) SSL certificate

The most basic of the three types of SSL certificates, the DV or domain validation certificate confirms the ownership of a domain through an automated online process. All you have to do is to complete a few basic steps to prove that you are the legitimate owner of the domain to get this certificate. Though easiest to get, this type of certificate helps in building only a basic level of trust.

After completing the DV certification procedure, you are allocated trust-building visual elements -- a static seal, a padlock icon, and an HTTPS prefix in your URL.

Cheapest prices and quick, automated processes are the two major benefits of a DV certificate. On the flip side, you get only the basic trust signs that aren't enough to satisfy the security concerns of more demanding customers or visitors.

Important to know: DV misses the most important trust sign — vetting the real, legal business that owns the domain name. For instance, suppose Mr X buys a domain 'abc.com' to lure gullible people into investing money in a business that seems genuine. Being a legitimate domain owner, he can get a DV through an automated process. With HTTPS URLs and other trust signs, it is easy to gain people's trust too. However, the company (or fraud mechanism) behind this domain isn't vetted/verified, which makes it easier for Mr X to continue duping investors of their hard-earned money.

Is DV SSL the right choice for you?: DV is best for a small and general website, which doesn't demand or require any sensitive information that can be misused like credit card numbers, social security information, or details of a financial portfolio. For websites that post children's stories, general e-magazines, personal blogs, professional portfolios, and other static websites on general topics, a DV certificate is a perfect and cheap solution. Securing more mission-critical sites that collect sensitive data like credit card numbers and social security details is a different story though.

Organisation validation (OV) SSL certificate

As the name suggests, the OV certificate not only validates the domain ownership but also vets the real identity of the organisation to which the domain officially belongs. It builds an extra layer of trust for visitors who have stronger concerns, by displaying the key information about the business. Interested visitors just need to check the details of the certificate to ascertain the credibility of the company.

As compared to a DV certificate, an OV certification requires a more detailed process carried out by an authorised certificate authority that demands and vets key documents representing the legal status of the company. Hence, along with domain ownership, this certificate also assures that it is owned by a legitimate organisation.

Though the issuance process is more expensive and demanding, this type of certificate does empower you to position your company as a legal, real company doing legitimate business. It makes you stand out and even more concerned visitors are able to trust your organisation.

Is OV the right choice for you?: Sites that ask for sensitive data that can be misused by threat actors should obtain this certificate. E-commerce sites with online payment gateways, digital health practitioners, government websites that demand citizens' information, defence-related websites, online trading platforms, or professional networking sites are all the right candidates for obtaining an OV certificate.

Extended validation (EV) SSL certificate

In the digital domain, visuals matter a lot. A company seal, security icons, or even assuring colours (like green) can go a long way in leaving a lasting impression, winning you a big deal or helping you forge business relations with dynamic brands.

The vetting procedure for issuing an EV SSL certificate is quite stringent and involves a manual process as well. It starts by verifying the ownership of the domain. After that, the certification authority asks for your identification number. Next, it gets your legal working contact number from relevant authentic sources. That contact number is manually verified by calling your office and talking with the real person. Only after satisfying all such verification parameters can you get the EV certificate.

Unlike an OV certificate, you don't just get a static site seal with a basic look, but get dynamic seals as well. Additionally, the full legal company name is displayed in the address bar along with a green-coloured padlock. Added to this, the entire address bar turns green as soon as your site loads. All these visible signs vouch for the legality of your firm, build a 'visual comfort zone' and reaffirm the credibility of your organisation.

Such certificates satisfy the sophisticated digital security vetting parameters of global brands and corporate conglomerates. If highly

expensive items like gold and diamond jewellery are being sold on a website, then such superior trust factors satisfy the higher trust demands of buyers. Though this certificate is the most expensive among all the SSL certificates, it is worth investing in the extra dollars as it has the potential to significantly boost your sales revenue.

Scope of different SSL certificates

Apart from the type of certificate, the other crucial question is: how many digital properties you want your certificate to cover? If you own a single domain property and do not want to expand in the foreseeable future, then a single DV would do for you. But what if you own, say, 20 different domains — most of which deal with e-commerce or collect sensitive client information? It wouldn't be practical to buy the more expensive EV certificate for each of these domains. Here is some guidance on that.

Wildcard SSL certificate

With a single wildcard SSL certificate you can protect the main domain and practically unlimited sub-domains related to it. For instance, *yoursite.com* (the main domain) may have three subdomains:

- *mail.yoursite.com*
- *login.yoursite.com*
- *ftp.yoursite.com*

This certificate relieves you from the stress and expenses of purchasing a separate certificate for every domain after going through the complete validation process, followed by an installation process for each. It also saves you a lot of time on repetitive processes and almost eliminates potential errors.

Note: Both DV and OV offer wildcard certificate options.

Multi-domain (or SAN SSL) certificate

One level above the wildcard SSL certificate is the multi-domain certificate, which helps to secure primary domains and their related subdomains. It does everything that a wildcard certificate can do, and more. If you own multiple domains and want a uniform and standard SSL security for all, then a multi-domain certificate is the right choice for you.

Is SSL certification only about visual trust signs?

Obviously, when you spend significant money on an SSL certificate you would like to get more than security icons or trust signals. Well, your certification authority does offer a specific amount of warranty or a payback if your customers become victims of a fraud. The amount of this warranty depends upon the type of certificate and how much it costs.

How to choose the ideal certificate authority (CA) for SSL

The next question is: who can judge the judge? It is you. Carefully consider some of the prime factors while finding the right SSL certificate provider/CA for your site. Before buying any certificate, thoroughly vet the reputation, credentials and experience of the certificate authority.

Also, ask questions. Do they have a credible history? What type of customers do they have in their repertoire? Do they have an impressive portfolio of regular customers? Most importantly, is the company passively following old industry standards, or is it actively investing in research and development on how to prevent the latest cyber frauds? All such questions will help you to make informed decisions and get the best value for your money.

Also make another very important check: Has any major browser banned the CA? The very objective of the SSL certificate is defeated if the CA has been banned by a major browser.

How long does it take for an SSL certificate to be issued?

The time taken to issue a certificate varies and depends upon the validation procedure and its requirements. A DV certificate, for instance, is issued within minutes as it has the least verification requirements. An OV SSL, with more detailed vetting requirements, can take up to three days for issuance. Since it has the most demanding vetting process, the EV certificate can take up to four days for issuance.

The validation period of the certificate, along with its credibility and scope, plays an important role in influencing its price. It is always best to use your discretion, and find a fine balance between the price and value of the certificate.

A few reputed SSL certificates go the extra mile and also offer extra security measures to customers. You can never be secure enough on a digital platform. So it is always best to see if such additional security elements will be helpful. However, your prime focus should be the credibility and portfolio of the company. It isn't wise to compromise with that just to get some extra security elements.

SSL certificates encrypt the data and information that customers share with organisations. They help to save customers from data theft and misuse by threat actors. But it's always advisable to check the credibility and reviews of a certificate authority carefully before buying a SSL certificate from it. END

By: Jitendra Bhojwani

The author is an open source enthusiast and a technical blogger.

A Quick Look at Cloud Cost Management Tools

Whether you are using a single cloud computing service provider or operating in a multi-cloud or hybrid cloud computing environment, you may be paying for resources that your organisation does not use at all. That is why cloud cost management is so important.



If you are a developer or software engineer, you will be using a range of cloud resources. Managing cloud costs, and knowing exactly where and why the expenditure on the cloud is happening is a huge challenge. If cloud-native technologies like microservices, containers and Kubernetes are being used, cloud utilisation may not be fully visible and the organisation may end up paying for far more services than have been used.

What is cloud cost management?

Cloud cost management is defined as the efficient monitoring, measuring and control of cloud costs. Traditionally, cloud cost management focused on cutting down cloud cost wastages in terms of under-utilised and forgotten resources but, nowadays, it focuses on architectural optimisation.

The advantages of cloud cost management are:

- High accuracy in the planning, budgeting and forecasting of cloud services
- Re-architecting resources for more profit
- Reducing operational costs and decommissioning wasteful services
- Effective load balancing management and better usage of computing resources

Cloud native cost management services

AWS Cloud Economics Center

Though the key driver for any enterprise to migrate to a cloud platform is better total cost of ownership (TCO), the benefits don't stop there today. Cloud economics or FinOps is the term used for creating a framework that can help to derive the expenditure in cloud adoption.

When a business is growing, good planning, budget control and predicting the pattern of cloud asset usage are important. Cloud economics addresses this space. There are many third party tools like CloudHealth, Densify or Apptio Cloudability available as cost management tools, which come with financial management, advisory and advanced analytics. Most of these tools work on any cloud platform — Azure, AWS or GCP.

At the same time, there are native tools available as well for cost management like the AWS Economics Center or Cost Management console.

AWS Economics is based on a cloud value framework and has the following benefits.

- *Cost savings:* Offers usage details on the cloud assets and TCO calculation to compare with existing and forecasted spends.
- *Staff productivity*: Improved automation tasks like DevOps pipeline, automated code review, and automated cost optimisation alerts reduce manual efforts.
- *Operational resilience:* A well-architected framework is used to ensure high availability, security and compliance, leading to highly resilient cloud platform usage.
- *Business agility:* Reduces errors in cloud deployment by using best practices and AWS partner solutions.

Azure cloud cost management

A well-architected framework has five pillars of cloud architecture development — cost optimisation, performance efficiency, operational excellence, reliability and security. Of these, cost optimisation will give immediate to long-term benefits, and requires a cost adoption framework and operating model to build an efficient cost pillar.

Cloud cost management includes cost transparency and cost governance as per the FinOps framework guidance. This can be implemented by using the Crawl-Walk-Run principle, where we start with cost transparency and the operating model (crawl), move towards cost optimisation and automation (walk) and, finally, implement cost management solutions and governance policies for efficient cost optimisation (run).

The Azure platform calls this the Build-Measure-Learn principle (similar to the Crawl-Walk-Run principle). It suggests reviewing the cost principles, including right-sizing of resources and resource tagging, and then developing a cost model to decide the billing models for resources. These will capture cost requirements, policies for infrastructure provisioning and associated budget allocation. Cost alerts and budget thresholds should also be set. Resources should be tagged so that priorities can be set in cost allocation for production and non-production environments, to create a workflow process in budget management.



Figure 1: Cost per customer report

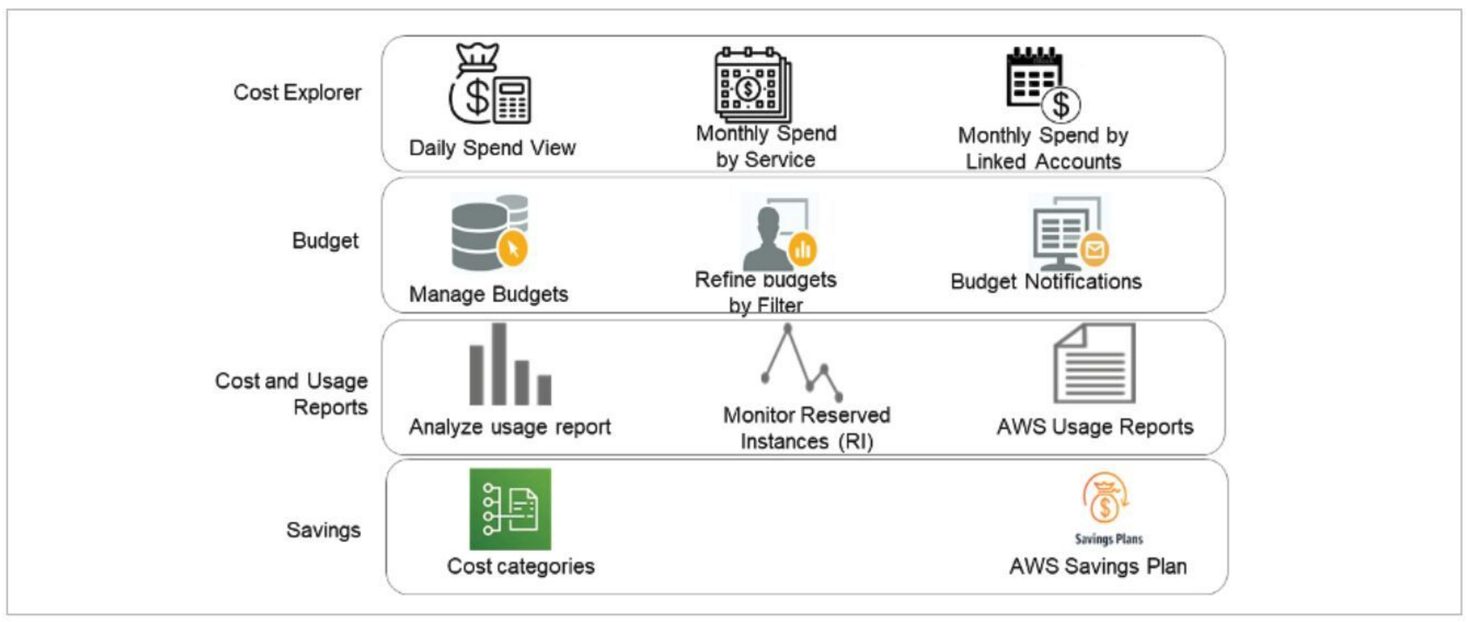


Figure 2: Amazon cost management solutions



Figure 3: AWS cost management dashboard

Google cost visibility services

The Google Cloud Platform or GCP cost management suite is built on three tools.

- *Cost visibility:* Provides the cost per resource.
- *Resource usage optimisation:* The cost advisory facility helps you to tune the cost per resource by giving

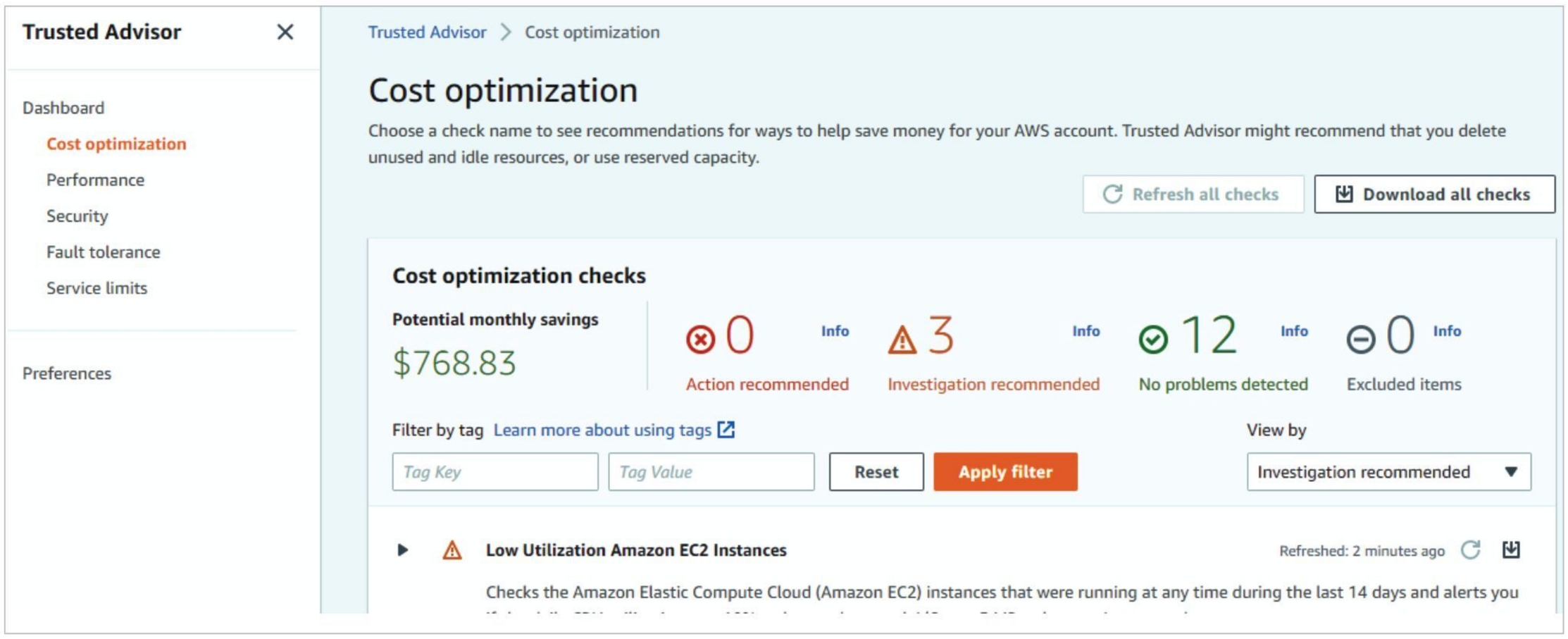


Figure 4: AWS Trusted Advisor

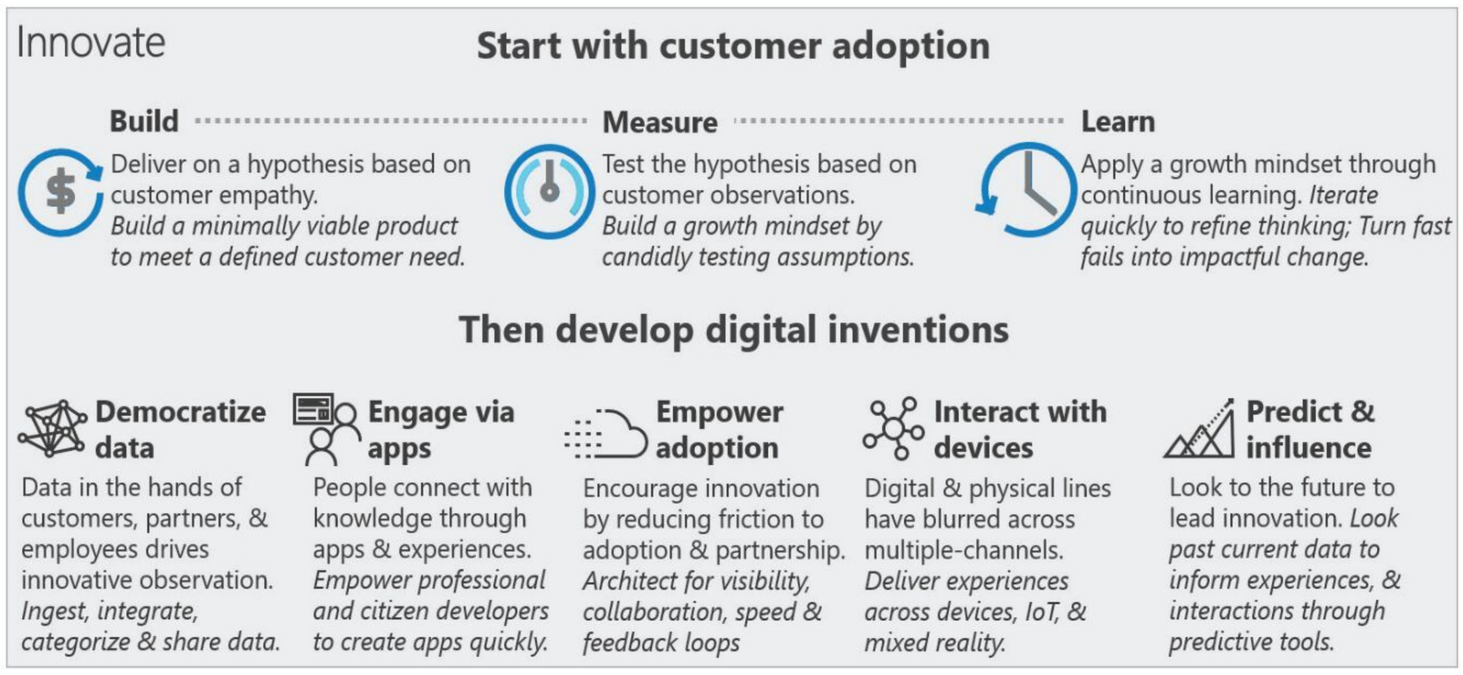


Figure 5: Azure guidance principles for cost optimisation (*Image source:* Azure documentation)

the history of and the prediction for cloud service usage.

- *Pricing efficiency:* Advisory guidelines are given to optimise the cloud services, like use of cost-efficient compute or storage resources based on usage patterns.

Cost management tools provide dashboard visualisation of resource usage. Alerts are given with respect to over usage or under usage of services, and a usage summary can be scheduled (weekly, monthly or quarterly). Billing

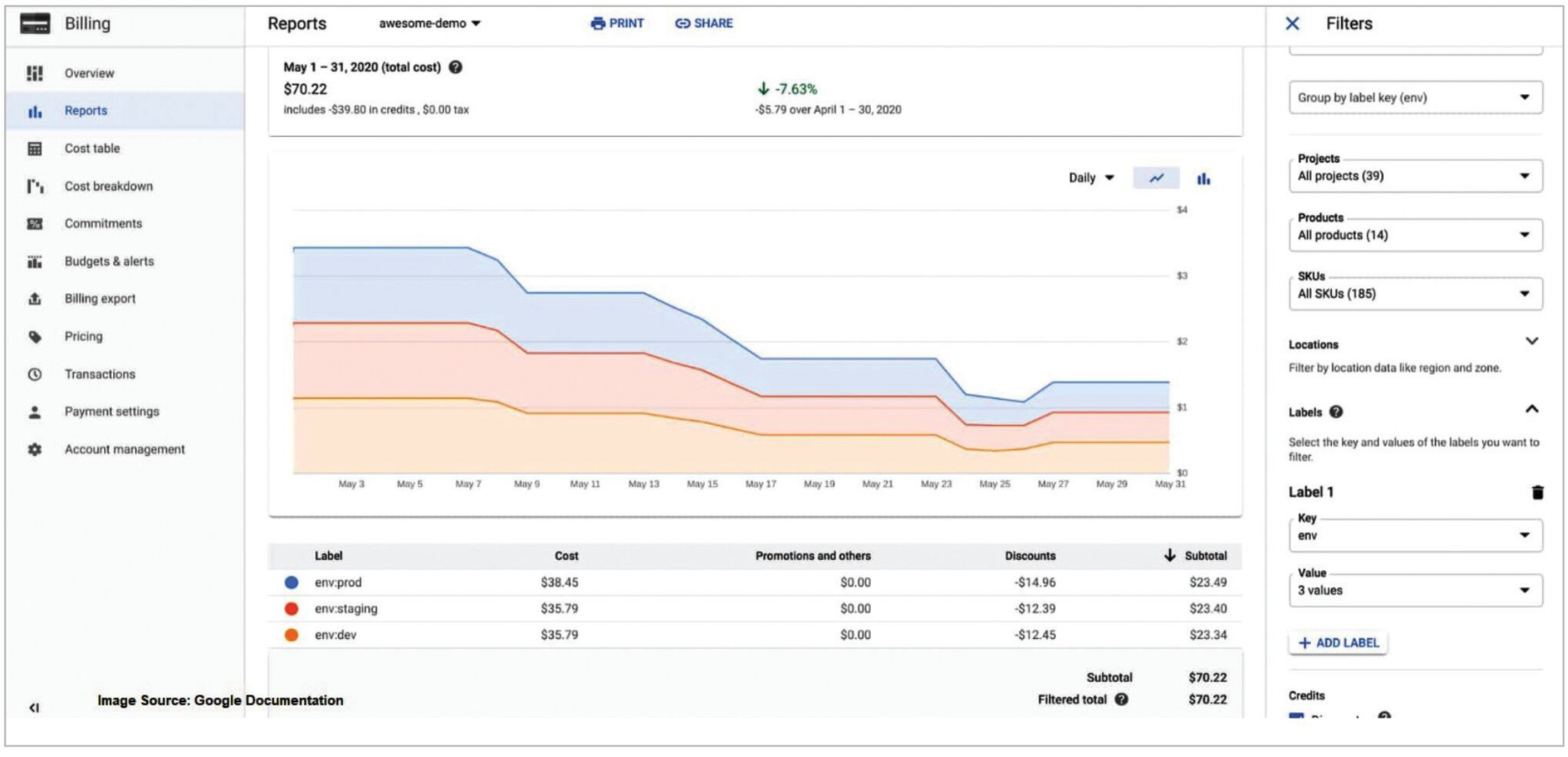


Figure 6: GCP cost summary dashboard (*Image source:* Google documentation)

exports to CSV or JSON files can be used for historical analysis.

Google Cloud Platform (GCP) also has many native cloud services for cloud billing; it has reporting, forecasting and optimisation features for compute and storage services, some of which are listed below.

- Cost management provides visualisation reports on service utilisation, filtered by labels. Data Studio can be used to build custom dashboards.
- Billing APIs and data collection APIs (StackDriver, Resource Manager API) offer integration with native and third party cost management tools.
- Google Recommender can be used for optimising cost and usage, based on patterns of resource usage. Resource hierarchy can be used for fine grained resource management for cost allocation.
- Quota limits can be used to proactively control the spend rate on resources including apps and infrastructure.
- Budgets and alerts can throttle cap costs to control spends. These can be used to closely monitor costs, and send alerts through an SMS or e-mail using programmatic budget notifications.
- GCP can be integrated with 'Cloud Build', 'Cloud Pub/Sub' (event/alerts) and cloud functions for cost reporting, billing and notifications.

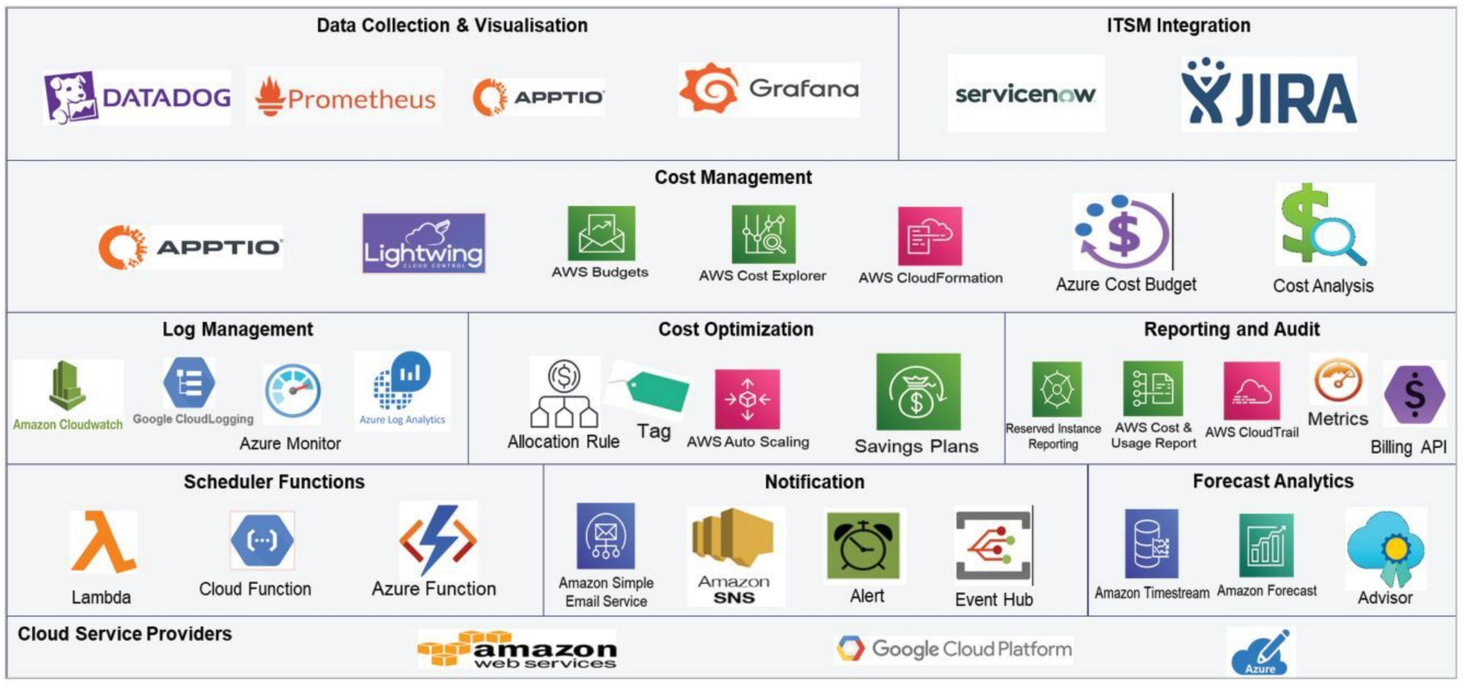


Figure 7: Integrated FinOps toolchain ecosystem

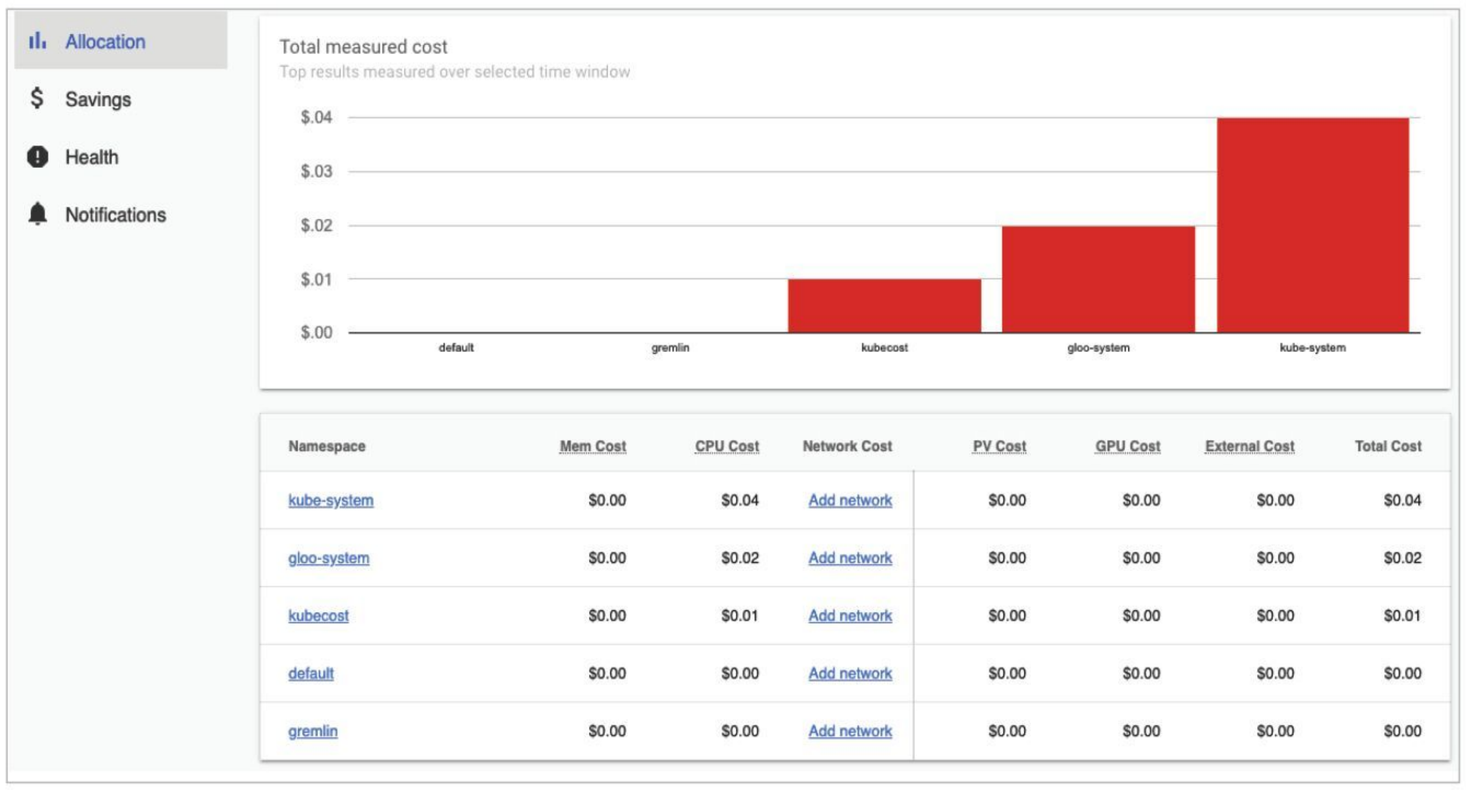


Figure 8: Kubecost dashboard (*Image source:* Azure documentation)

Some best practices to handle cloud costs efficiently

Cost management and cost transparency are the two common principles of cloud governance. The FinOps framework addresses both these principles.

The top seven ways to optimise cloud cost usage are listed below.

- *Shutdown unused resources/instances:* A common issue in resource optimisation is handling unused resources or instances, particularly in non-production environments. Monitoring and controlling these will enable better cost optimisation.
- *Right-size underused resources:* During cloud migration, infrastructure sizing is done based on the existing on-premises infrastructure. We may not know the performance requirements (completely) at the initial stage, and hence right-sizing resources like VM instances, storage services and database size can help reduce costs.
- *Reserve instances or spot instances for consistent long-term workloads:* When moving to the cloud, we generally have a clear idea of what we need for long-term usage like data lakes or FTP landing zone. For these services, using reserved or spot instances for three to five years can give as much as 50 per cent cost benefit.
- *Choose the hybrid cloud approach for reducing migration costs:* Instead of moving all applications at once to the cloud, we can migrate in stages, keeping some key applications on-premises.
- *Use auto-scaling features for the required resources*: Many cloud services have the auto-scale feature, which helps to scale-up computing and storage when there is higher usage and scale these down in case of lower usage. This feature too helps to reduce costs.
- *Budget your resources*: We can set a quota for resources using a budget and cost allocation, so that if there is dynamic burst in resource requirement it can be controlled well within the allocated budget by enabling alerts and notifications.
- *Choose the right compute services for better performance and costs:* Though cloud service providers like

Azure, AWS and GCP have multiple compute models and instance types, we may not know if we are using the right compute size and model. If we can optimise the compute model considering performance and technical requirements, we can manage costs better.

Open source cloud cost management tools

Kube-Downscaler

Kube-Downscaler is an open source framework written in Python, which can be used to handle automatic shutdowns or set the uptime for Kubernetes worker pods based on usage patterns (for example, shutdown during the weekend and set uptime during office hours).

This framework runs on a single pod in the Kubernetes cluster and can work with Azure (AKS), AWS (EKS) and GCP (GKE). It has a configuration setup that can be given for the shutdown time period or uptime running period. It can be scheduled to run once or regularly for a given period (monthly, yearly).

Alternatively, we can use Kubernetes-Ops, which is another open source framework for handling end-to-end Kubernetes clusters. It can create pods/clusters, and manage, deploy, scale and monitor them in a single pane of glass.

Kubecost

Though cloud cost governance aims at managing compute and storage services to a large extent, managing container costs is also a challenging exercise.

Kubecost helps to get monthly spends, costs for name space level, and deployment resource costs; it identifies cost efficiency in a Kubernetes clustered environment. It comes in both a free-to-use open source model (where there are limitations to using it such as a single cluster, metric retention period) and a commercial model.

We can get details of underutilised or over-provisioned resources (nodes and pods, and abandoned resources) based on usage footprints. The major features of Kubecost are cost allocation across nodes and pods, unified cost monitoring with integrated dashboard, cost optimisation insights to identify under-provisioned or over-provisioned cluster resources, and cost alerts as well as notifications for anomalous behaviour in resource usage.

Kubecost can be used with Azure Kubernetes Service (AKS), Amazon's Elastic Kubernetes Service (EKS) and Fargate, Google Kubernetes Engine (GKE) and other Kubernetes platforms in private cloud environments.

Integrated FinOps toolchain solution

FinOps is more of a culture adoption and is driven by the three-dimensional approach for cloud cost management — people (cost administration, team structure, training), processes (cost levers, maturity models, metrics definitions) and tools (cost management services, third party tools, cost advisory services).

FinOps is a collaborative activity between the finance/procurement team, technical IT operations team, and business management (COO, CIO). It can be developed as an agnostic framework for implementation in a multi-cloud environment (public/private/hybrid). It can also be easily integrated with third party tools for flexible implementation. For example, developing reports and dashboards could be made flexible with this integration of third-party tools.

Future trends in FinOps adoption

The adoption of FinOps is maturing in most organisations that have substantial cloud consumption as they need processes and procedures to manage, monitor, allocate and forecast cloud spending. With the introduction of FinOps, organisations will have deeper insights into the resources available or reserved, which will promote appropriate procurement of cloud resources and avoid unnecessary spend in the first place.

Choosing between the right FinOps platform from cloud services providers (Azure Cost Management, AWS Cost Explorer and GCP Cost Tools), third party options or in-house development is not an easy task. Factors like variability in size and usage of cloud resources, fast-changing technology, and the advent of new cloud use cases complicate the process of automating cost optimisation. Building a governance culture and mindset is vital, but not always easy. END



References

[1] *https://blogs.gartner.com/lydia_leong/2021/11/24/the-cloud-budget-overrun-rainbow-of-flavors/*
[2] *https://www.kubecost.com/features*
[3] *https://opengov.com/article/using-kubernetes-downscaler-for-time-of-day-and-day-of-week-automation*
[4] *https://www.apptio.com/blog/where-finops-and-tbm-intersect/*
[5] *https://www.cloudtp.com/doppler/cloud-economics-getting-bigger-picture/*
[6] *https://www.softchoice.com/blogs/cloud-migration-adoption-management/whitepaper-finops-the-future-of-cloud-economics*

By: Dr Anand Nayyar and Dr Magesh Kasthuri

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

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.

Build a Smart Parking System for a Metro Station

This article will help you design a Web based application that automates a smart parking system for cars in a metro station using Node-RED.



A Web application is software that gets executed on a Web server. Every Web application is accessed by the end user through a Web browser. These Web applications are programmed using a client-server architecture, where the user (client) is provided services through a remotely located server that might be hosted by a third-party. A Web API (application programming interface) is available across the Web and can be accessed by the user using the HTTP protocol, as shown in Figure 1.

This article will demonstrate how to design a Web based application for an automated smart parking system for cars in a metro station. It is designed using open source Node-RED. This system creates an interactive and stylish user login form using a template node, where HTML and CSS are coded to get car owner details to automate the parking system. We can see the login and submission form flow diagrams in Figures 2 and 3.

The nodes used are as follows:

Http in node	http in
Function node	function
Template node	template
Http response node	http response
JSON node	json
Debug node	debug
Comment node	comment

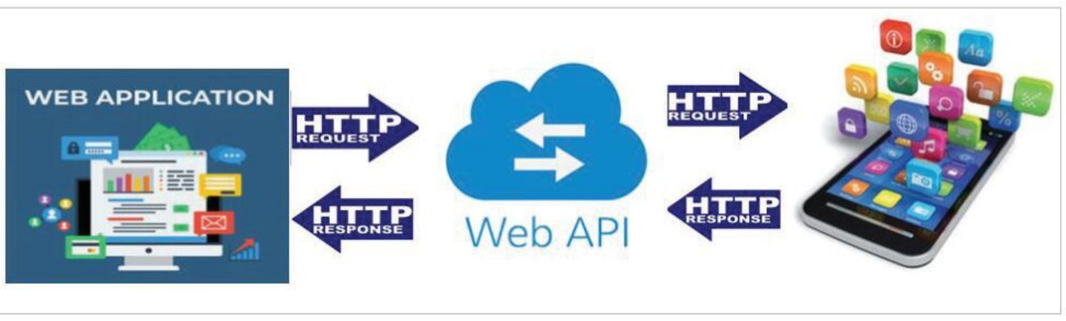


Figure 1: Web API

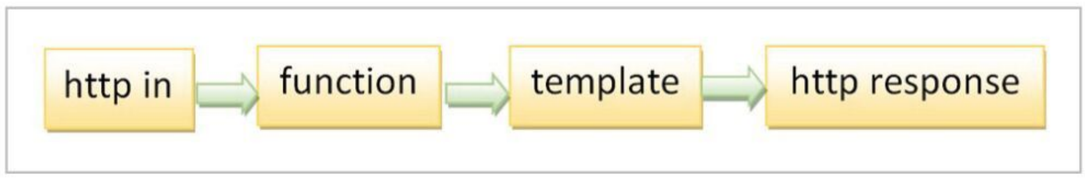


Figure 2: Login form flow diagram

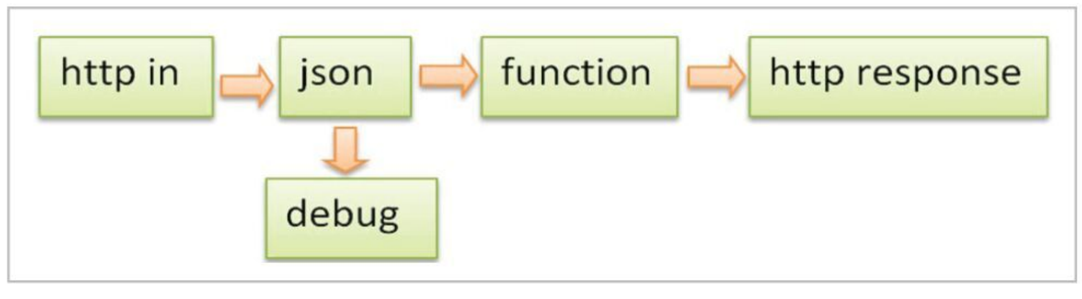


Figure 3: Submission form flow diagram

Metro smart parking node flow design

Node-RED is triggered using the command 'node-red'. Through the URL *http://127.0.0.1:1880/*, the node-RED UI flow browser is enabled. We have considered that the Node-RED setup is done and working.

Follow the steps given below to create the login and submission forms.

Login form

1) From the node palette, drag and drop *http* in node, which creates an HTTP end point for creating Web services.
2) Connect *http in* node to *function* node. The latter helps to code JavaScript functions to run against the messages being received by the node.
3) Connect *function* node to *template* node, where the latter creates a Web API based on the provided template.
4) Connect *template* node to *http response* node; the latter sends responses back to requests received from an *http input* node.

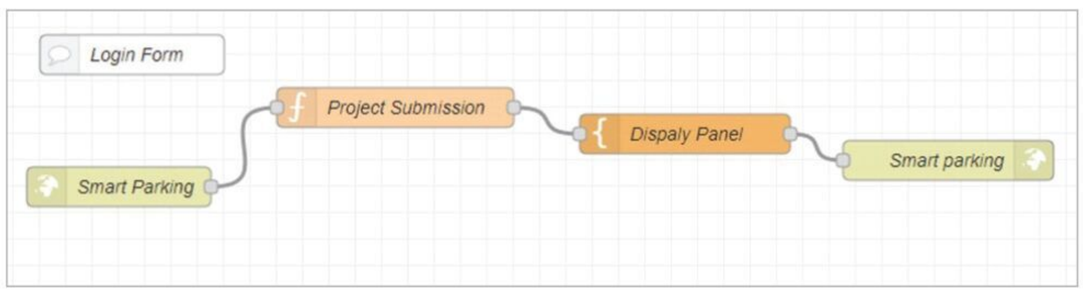


Figure 4: Login form for smart parking for cars

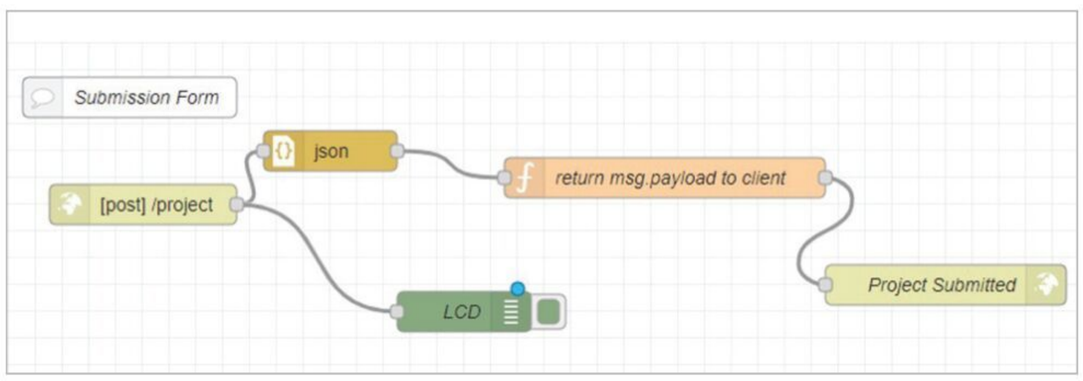


Figure 5: Submission form for smart parking for cars

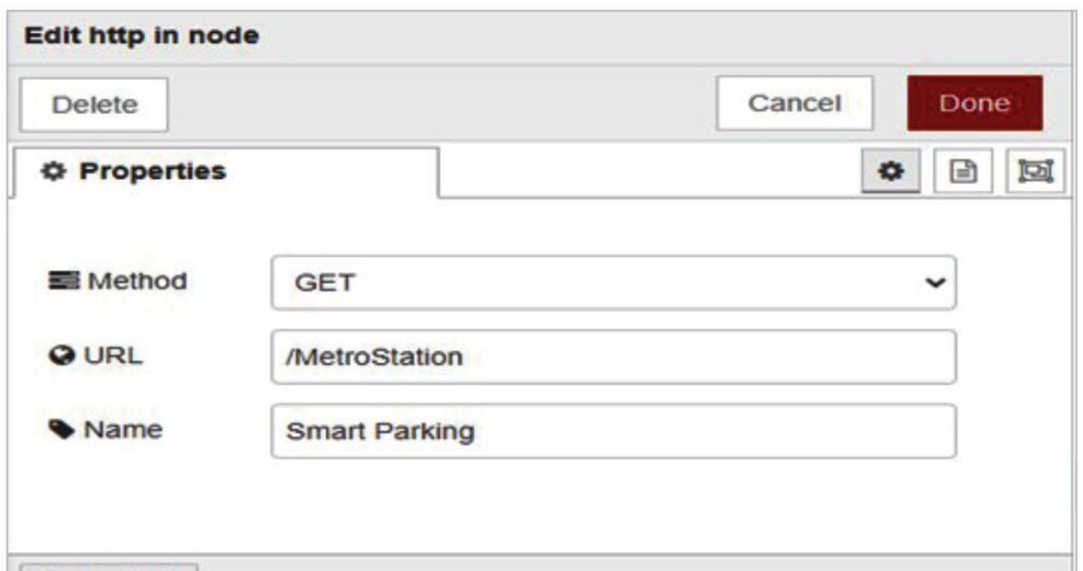


Figure 6: *Http in* node property configurations

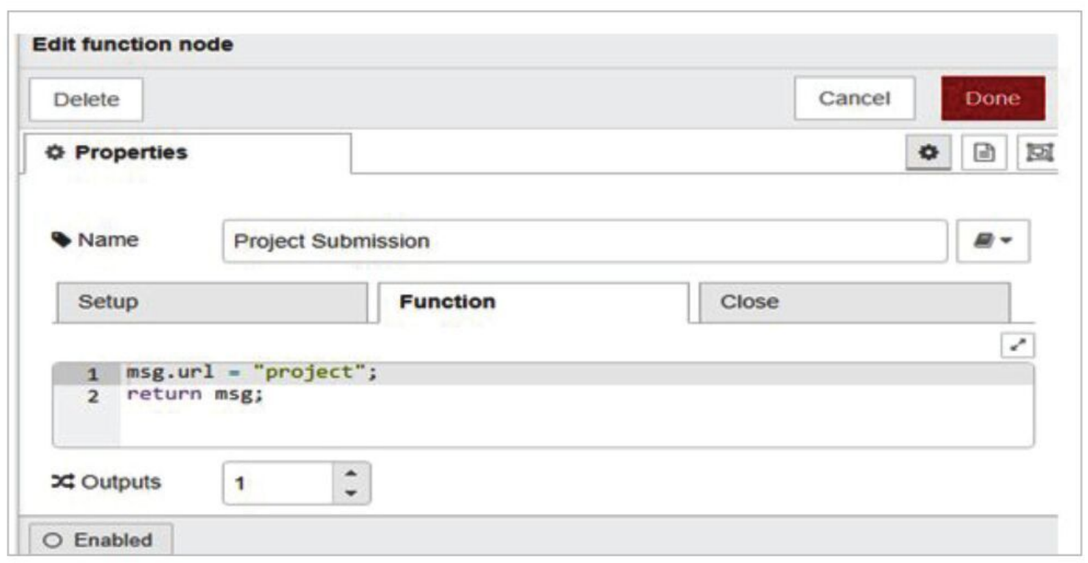


Figure 7: *Function* node property configurations

Submission form

1) Drag and drop *http in* node and connect it to *json* node, which converts and communicates the data as JSON string.
2) Connect *http in* node to *debug* node, which gives output

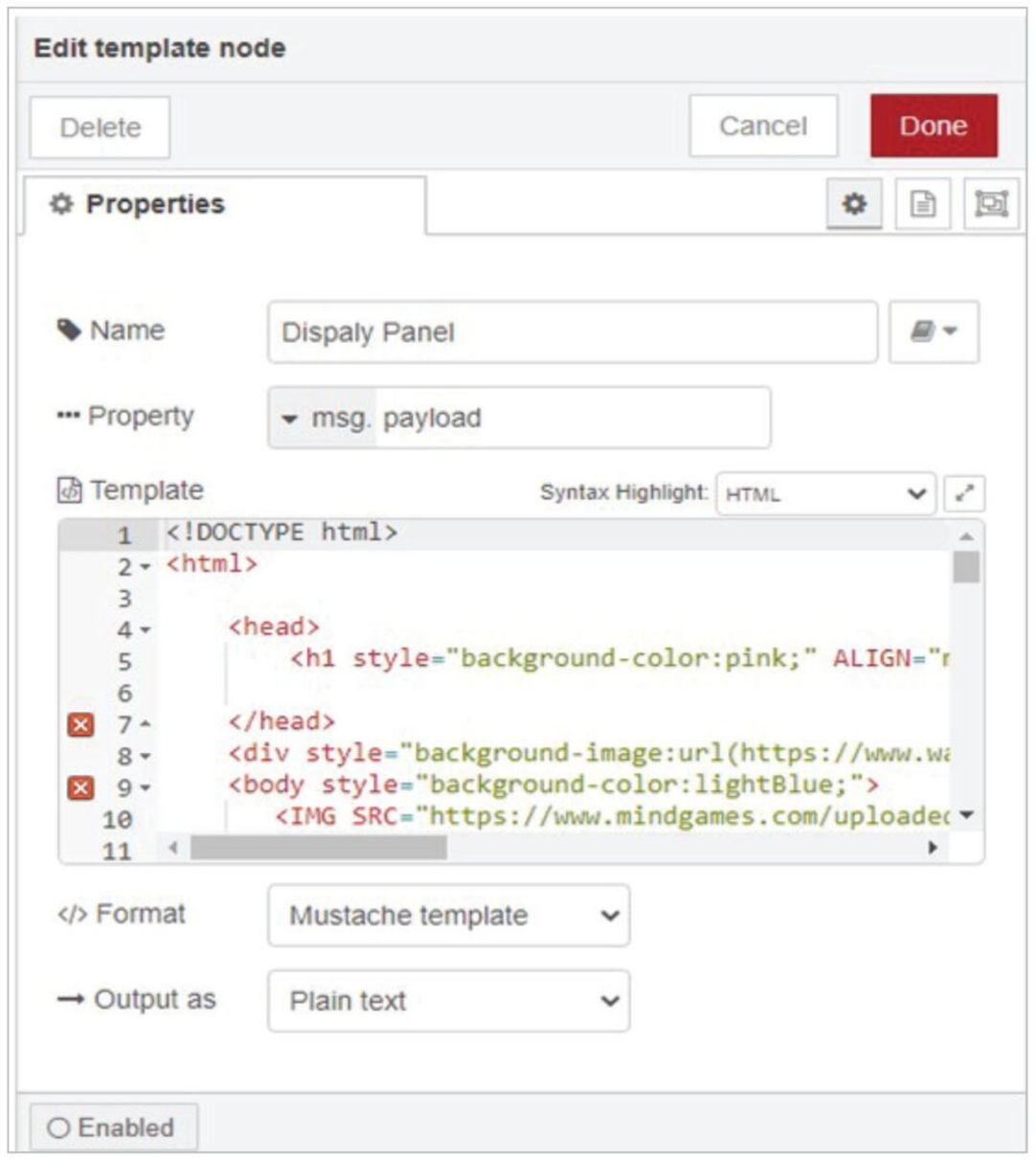


Figure 8: *Template* node property configurations

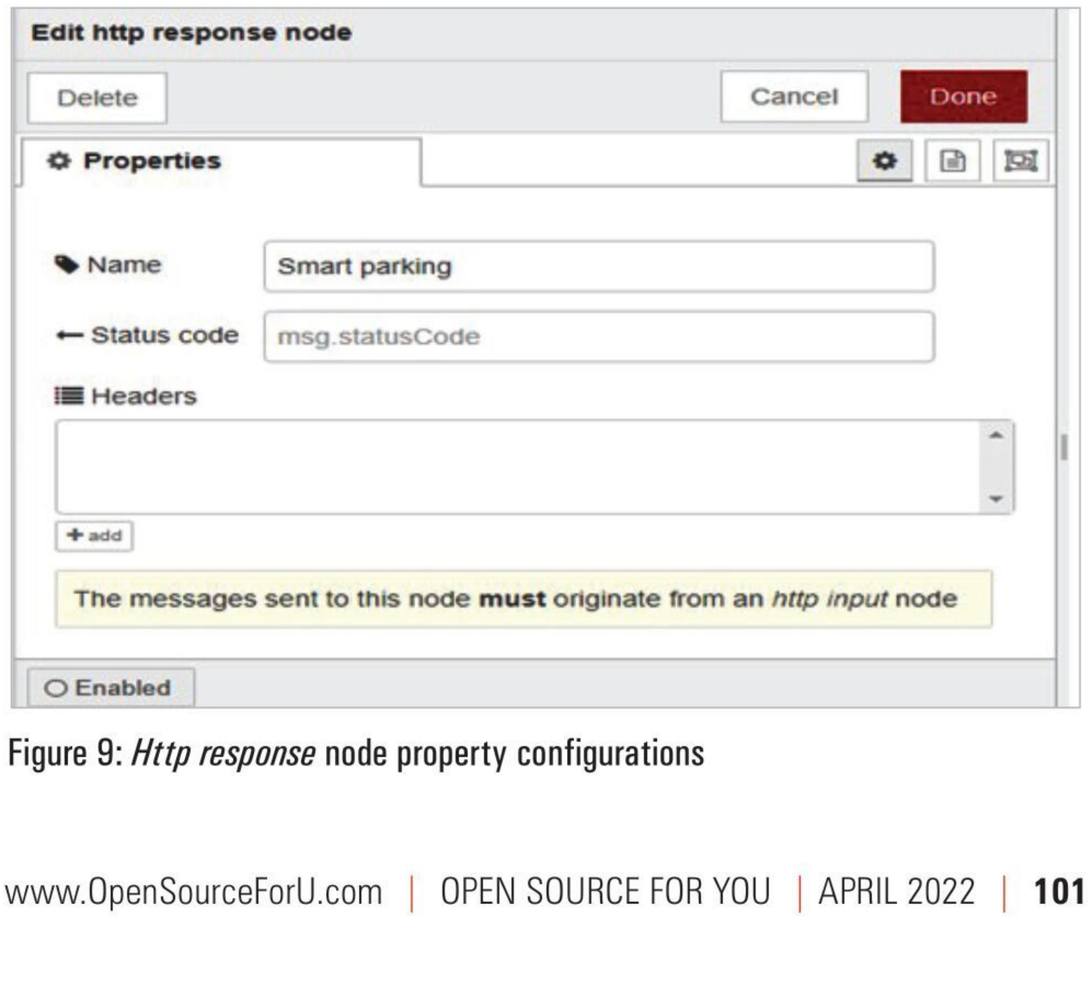


Figure 9: *Http response* node property configurations

in a debug monitor.

3) Place and connect *json* node to *function* node and connect the latter to *http response* node.

After the creation of a complete flow, click on the *Deploy* button in the top right corner of the Node-RED window. To view the user interface, go to the link *127.0.0.1:1880/ui/*.

Once you enter and then click *Submit*, it will take you to the next page where you can read all the news articles.

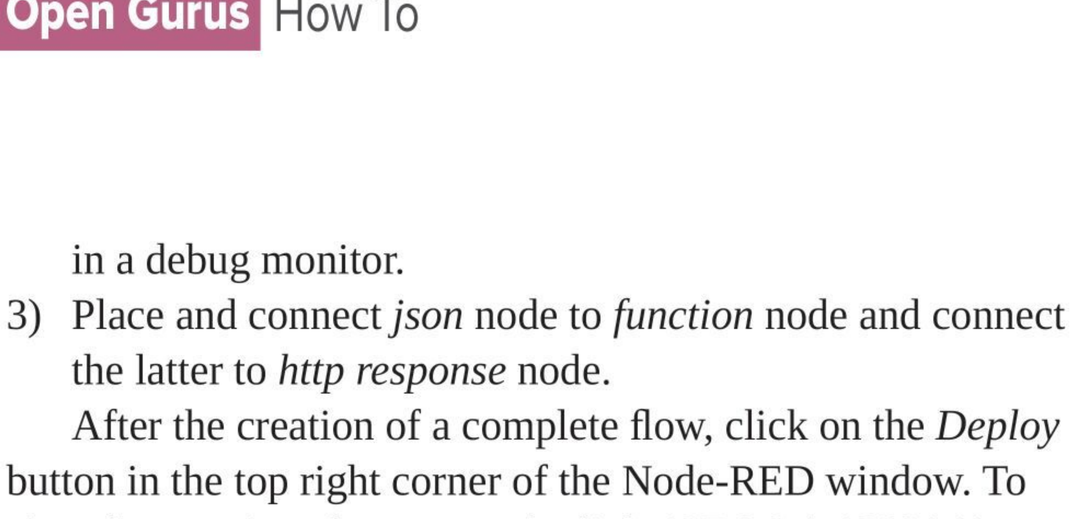


Figure 10: *Http in* node property configurations

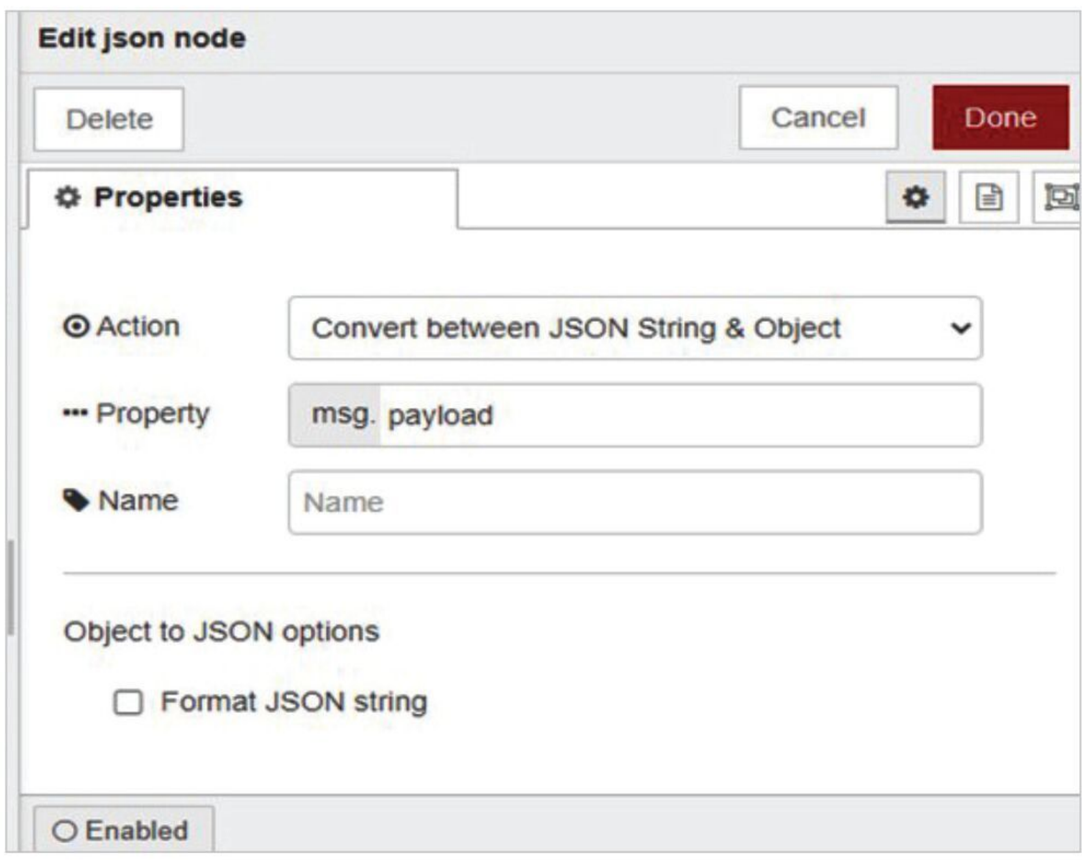


Figure 11: JSON node property configurations

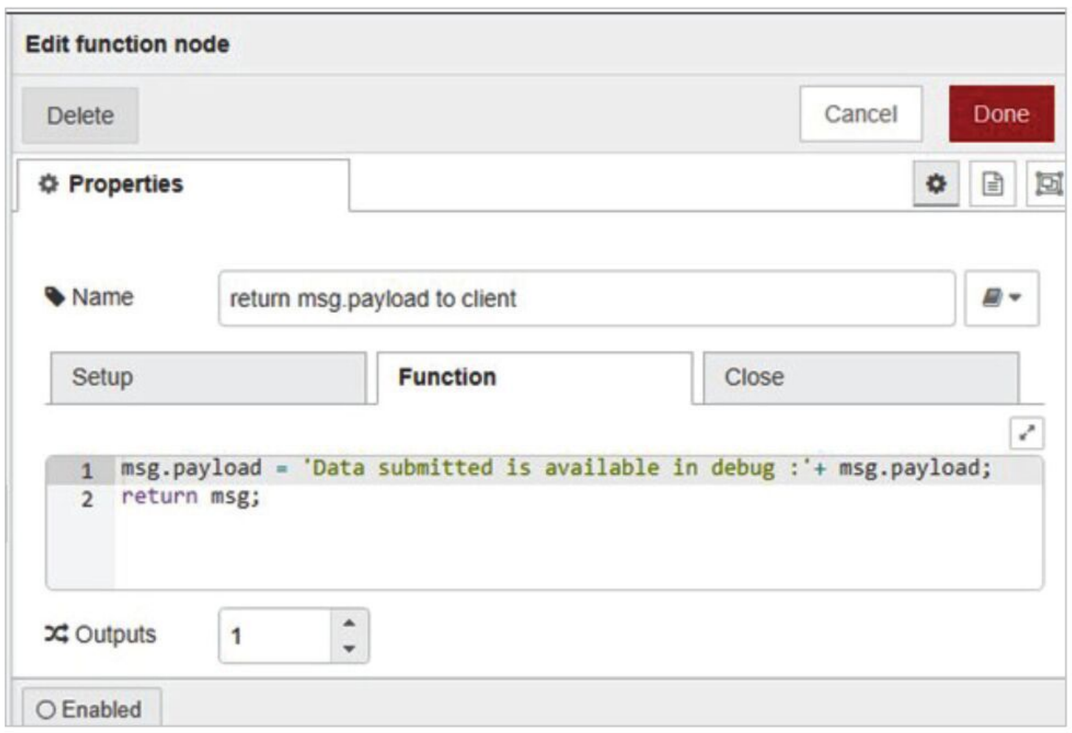


Figure 12: *Function* node property configurations

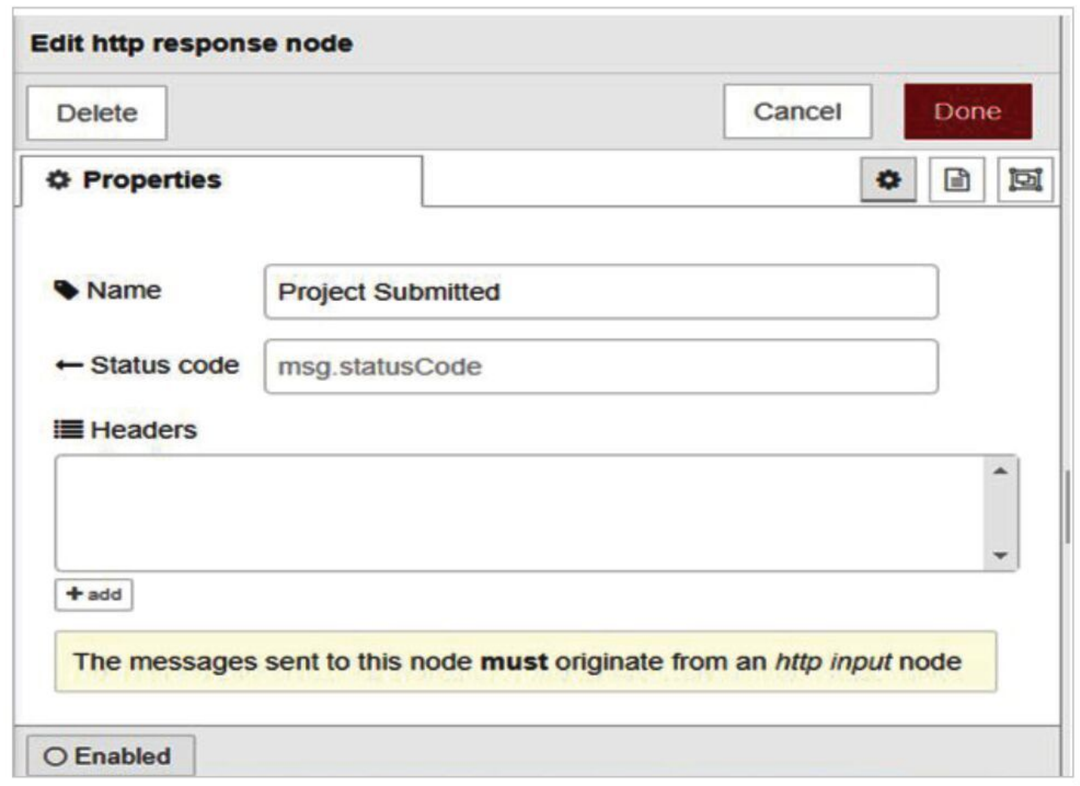


Figure 13: *Http response* node property configurations

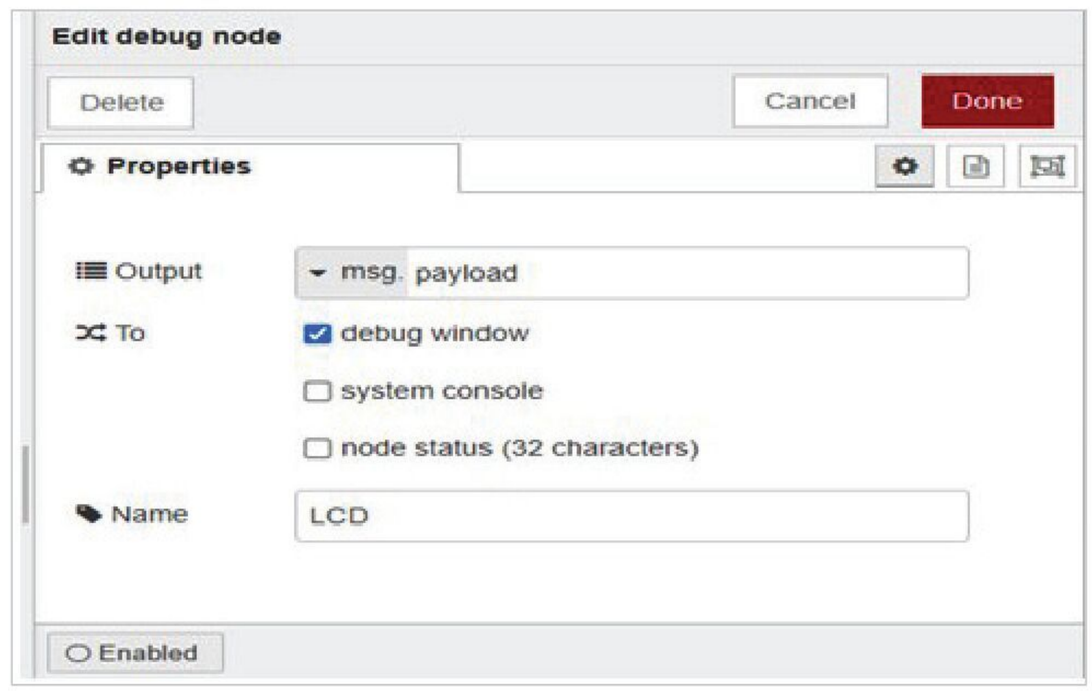


Figure 14: *Debug* node property configurations



Figure 15: User login UI

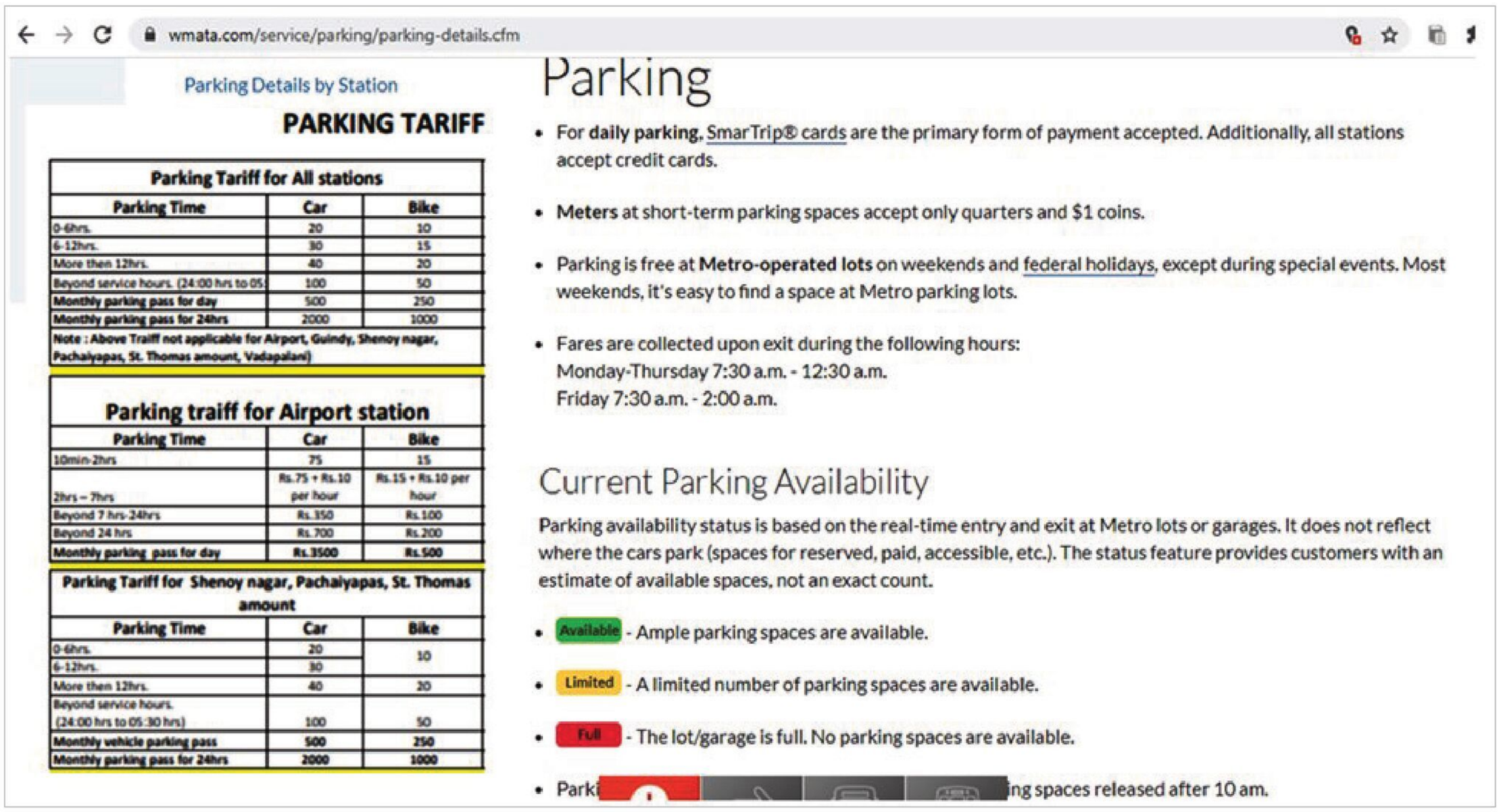


Figure 16: Metro parking tariff

Node-RED workflow

In a single flow of Node-RED, you can create both the login form and submission form, as shown in Figures 4 and 5.

Now we will configure the Node properties.

Login form: Edit the *http in* property by choosing the method 'Get' and set the URL to '/MetroStation' and configure the name as 'Smart Parking'.

Note: The URL can be any user defined local variable.

Now select the *function* node and edit its properties by coding the 'msg.url = project' and configure the name field with 'Project Submission'.

In the *Property* window of the *template* node, configure the appropriate HTML code required for the login form and specify the name as 'Display panel'. The Mustache template format is being used in this flow. Mustache is a simple Web template system that is described as a logicless template engine. It does not have any explicit control flow statements, such as 'if' and 'else' conditionals or 'for' loops. Looping and conditional evaluation can be achieved using section tags processing lists and lambdas.

Configure the edit property of *http response* node with the name 'Smart Parking' (Figure 9).

Submission form: In the *edit property* window of *http in* node, choose the method 'POST' and the URL '/project'.

In the JSON node edit window, set *Action* as 'Convert between JSON String & Object'. Refer to Figure 11.

The *function* node is configured as specified in Figure 12.

In *http response* node, edit the property name as 'Project Submitted'.

Note: Also add the *comment* node with comments as 'Login Form' and 'Submission Form'

Dashboard UI Web page

When the user clicks on *Submit*, the data given will be displayed in the UI and the *debug* node. If *Reset* is clicked, the details will be cleared, allowing the user to enter new details (Figure 15).

Metro parking rates are provided through a hyperlink, and the tariff output is displayed in the UI. So the smart parking for cars is automated with appropriate hyperlinks to exhibit the parking tariff at the metro station. The final outputs of this automated system are retrieved and displayed in the Node-RED dashboard UI and debug monitor. END

By: Dr Maheswari R.

The author is an associate professor at the Vellore Institute of Technology, Chennai.

Why Edge Computing is Important for IoT Applications

Smart IoT devices are all around us today, and are growing exponentially each year. With this growth come challenges in managing the data generated by these devices. Edge computing comes to the rescue here, as it processes data closer to the IoT device and helps to generate real-time responses.



IoT devices generate a very large amount of data that is processed and analysed for use in various applications. In the early days of IoT, most devices sent this data to the cloud for storage and analysis. As cloud servers are located very far from IoT devices, this led to a delay in getting processed data. New IoT applications require real-time data processing, for which cloud computing is too costly and not fast enough. Edge computing brings data processing services near the devices or the source of the data. This allows data from various IoT devices to be processed and analysed at the edge of the network, which results in lower latency and enhanced efficiency in data transportation. Data can then be sent to the cloud or data centre for long term processing and storage.

The Internet of Things or IoT refers to the large volume of intelligent devices connected to the Internet. These devices can be called nodes; they sense, collect and send data to each other through various communication protocols. IoT devices collect a large volume of heterogeneous data generated by sensors, which needs processing for extracting useful information to offer services to users.

In traditional computing, data gathered by IoT devices is sent to cloud servers for processing, and the results are transferred to IoT devices for necessary action. The time required to send and get processed data from the cloud server is more, which is not acceptable for real-time applications such as healthcare, smart transportation, and smart grids.

The challenges and emerging applications of IoT

As the data generated by various IoT applications increases exponentially, it brings new challenges in managing

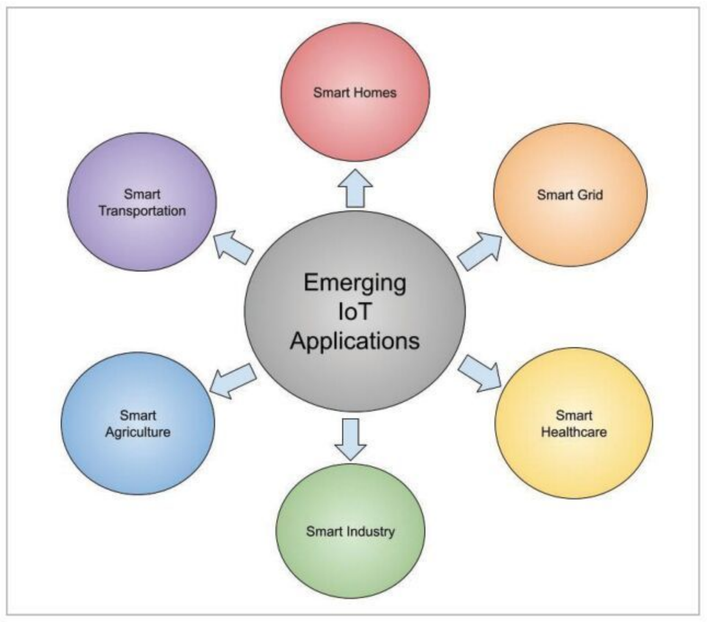


Figure 1: Emerging IoT applications

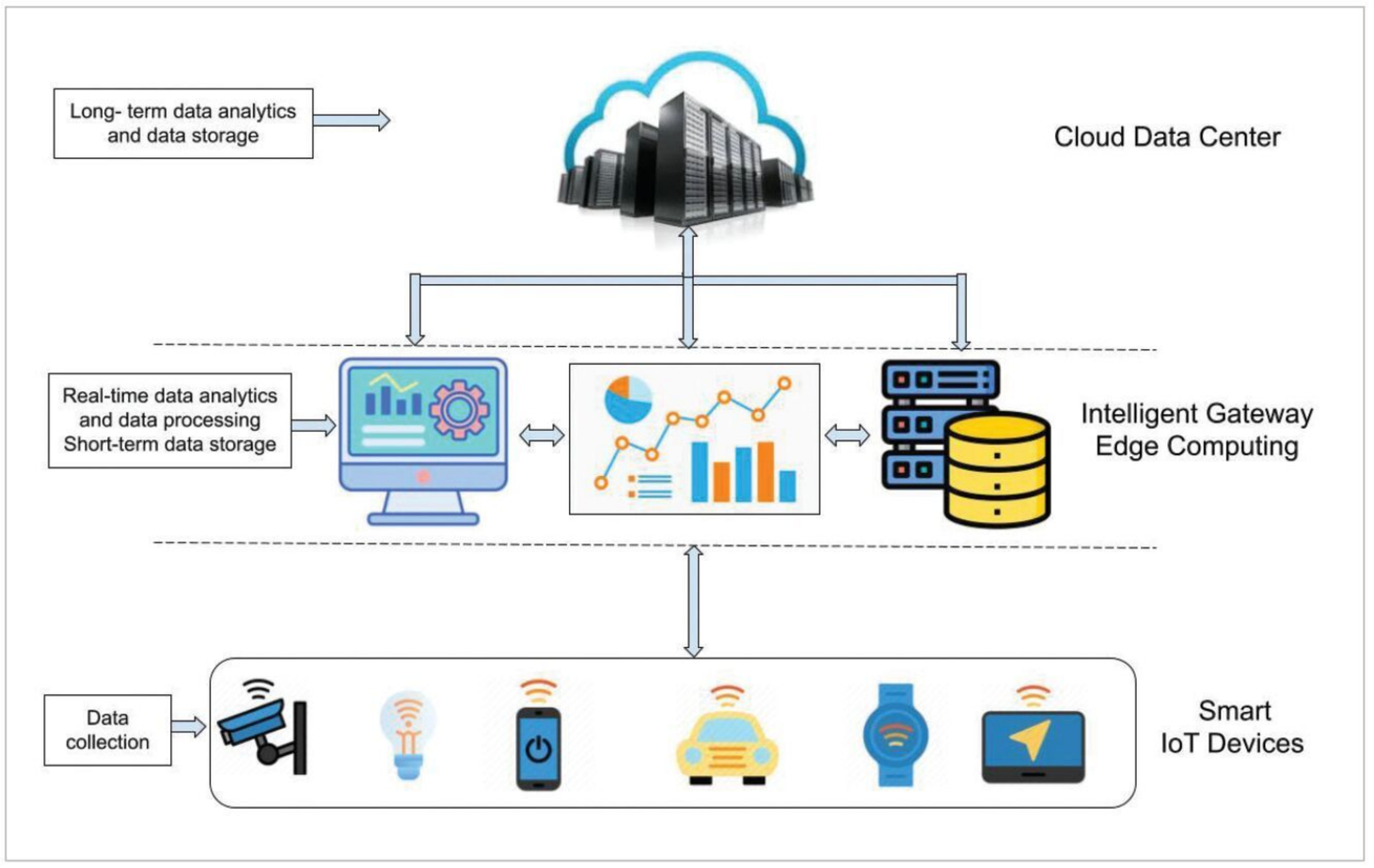


Figure 2: Edge computing-IoT layered architecture

it. In many IoT applications, where reliable and real-time data delivery is required, cloud computing based services are not suitable to meet their demands. Most IoT applications face the following challenges:

- Scalability
- Reliability
- Transmission delay
- Interoperability
- Availability
- Security and privacy

The rapid growth in IoT devices has opened up several applications in different domains, and especially in the field of communications. Many of these emerging applications are open to different sets of goals and requirements, but play a vital role in making our everyday activities easy, smart, and safe. Figure 1 shows the various emerging IoT applications.

Role of edge computing in IoT

Edge computing processes and stores the data received at the edge of the network, which is near to the end users. The processing of data closer to the end devices decreases the transmission time, lowers bandwidth needs and increases the battery life of IoT devices. Edge computing offers benefits for many IoT applications, including reduced response time and improved communications efficiency, as compared to cloud computing.

IoT applications can be classified with respect to their delay sensitivity — a particular application may or may not be sensitive to delay, depending on the type of parameters it manages. We can choose between cloud computing and edge computing based on this delay sensitivity.

Edge computing performs well in comparison with cloud computing for IoT, though it has limited computational power and storage. Most IoT applications require real-time services rather than greater computational power and storage.

Edge computing with IoT is extensively used in many industrial applications; it enables numerous new services for enterprises and users. Many emerging IoT applications, such as smart vehicles, smart grids, depend on edge computing. Advantages of edge computing in IoT include:

- Low latency
- Longer battery life for IoT devices
- Low bandwidth requirement
- Resilience in systems
- Scalability

Fusion of edge computing with IoT

Edge computing is a cost-effective framework for extending the functionalities of cloud computing. In real-time IoT applications, it reduces the latency and also the response time. The features of edge computing such as low latency, mobility provision, location awareness, and dense distribution can solve the challenges the IoT industry is facing.

Figure 2 shows the three-layer architecture of edge computing based IoT. The bottom layer contains various smart IoT devices, which generate the data. Data gathered from this layer is sent to the edge layer, where data processing and data analytics tasks are performed at the edge nodes to make real-time decisions. Based on the decisions taken at the edge layer, data is sent to cloud servers for exhaustive analysis and stored there.

Emulators and simulators for edge IoT computing

The excellence of any IoT system is usually evaluated by the degree to which it fulfils the requirements of its users. The performance of an IoT application can be measured with various quality metrics like latency, energy efficiency, and scalability. These measurement metrics vary from application to application, because every application has different requirements. The simulation of real-world IoT infrastructures has gained substantial interest in the research

community because performing experiments on them is both an expensive and time-consuming process.

There are a number of simulators that facilitate the evaluation of various aspects of IoT systems. Many of these are available for cloud computing and some of them are used to simulate the fusion of edge computing in an IoT environment. Here we briefly describe well-known edge computing simulators used for modelling, and compare them in terms of quality. Figure 3 shows the various simulators/emulators available for edge computing. Table 1 gives a comparison between these.

iFogSim simulator is used for simulating fog and edge computing, and to provide an evaluation tool to manage various resources. It is designed based on the CloudSim simulator, which is primarily used to simulate a cloud computing environment. It provides a quantitative evaluation of latency, energy utilisation, and network traffic.

IoTSim simulates edge computing scenarios, where a huge amount of data is processed in a Big Data processing system by the IoT system. It gives special focus on simulating communication and network delays among data storage and data processing virtual machines.

Table 1: A comparison of edge simulators/emulators

Attributes	iFogSim	IoTSim	Edge CloudSim	FogNet-Sim++	EmuFog	FogBed
Scalability	×	√	×	√	×	×
Mobility	×	×	×	√	×	×
Network delay	×	×	√	√	√	×
Energy consumption	√	×	×	√	×	×
Bandwidth utilisation	√	√	√	×	×	×

EdgeCloudSim is explicitly designed for the evaluation of edge computing scenarios in IoT. It provides various models such as the mobility module, edge server module, and network link module to assess the several features of edge computing. It is user-friendly and provides a method to configure devices and applications using XML files.

FogNetSim++ is based on the prominent simulator OMNET++, which contains widespread network libraries and provides numerous network protocols. It empowers researchers to integrate user-defined mobility models and manage delivery mechanisms.

EmuFog emulator supports repeatable and manageable experiments with real-life applications. It emulates network topologies and is designed on top of the MaxiNet network emulator. However, it does not support mobility, both for fog nodes and clients, and also doesn't provide hierarchical fog infrastructures.

Fogbed is an extension of the Mininet simulator. The network topology can be changed dynamically using Docker containers. However, it does not support the main facets of fog computing such as fault tolerance, scalability, security, and reliability.

IoT devices must be able to process data closer to end users, especially for time-critical applications. Edge computing is the best option for this. However, it's still in an early stage of development and more research is needed for the efficient integration of edge computing with IoT. END

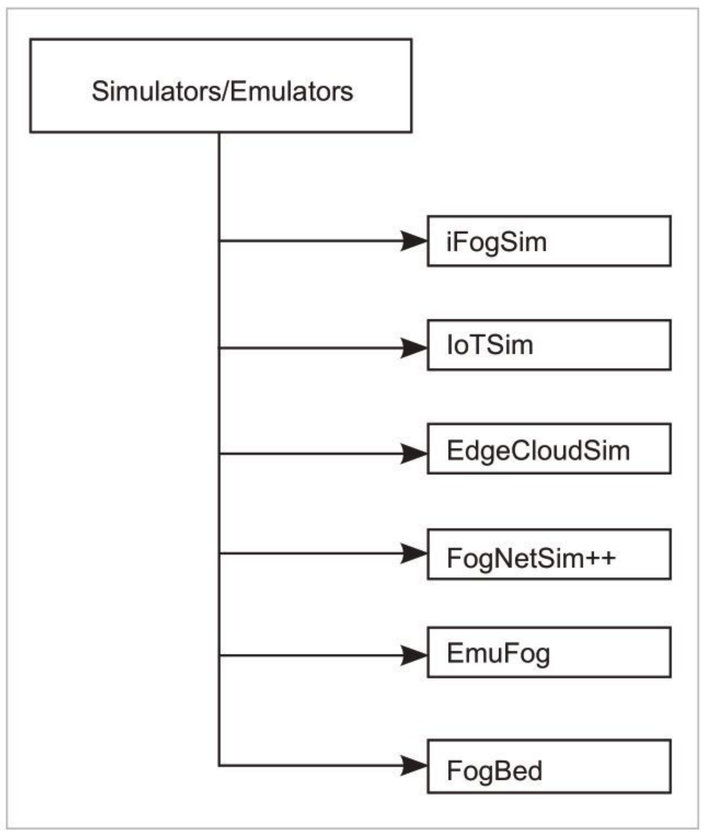


Figure 3: Simulators/emulators for edge computing

References

[1] Ashouri, M.; Lorig, F.; Davidsson, P.; Spalazzese, R. Edge Computing Simulators for IoT System Design: An Analysis of Qualities and Metrics. Future Internet 2019, 11, 235, *https://doi.org/10.3390/fi11110235*

[2] N. Hassan, S. Gillani, E. Ahmed, I. Yaqoob and M. Imran, "The Role of Edge Computing in Internet of Things," in IEEE Communications Magazine, vol. 56, no. 11, pp. 110-115, November 2018, doi: 10.1109/MCOM.2018.1700906.

[3] Kar, Hushmat Amin and Ghulam Mohammad Rather. "A Survey on Edge-Based Internet-of-Things." (2019).

[4] *https://www.gsma.com/iot/resources/edge-computing-iot-report*

By: Malaram Kumhar and Jitendra Bhatia

Malaram Kumhar works as assistant professor in the computer science and engineering department at Institute of Technology, Nirma University, Ahmedabad, Gujarat. He is keen to contribute to open source projects.

Jitendra Bhatia is an assistant professor in the computer engineering department at Vishwakarma Government Engineering College in Ahmedabad, Gujarat.

EFY GROUP
YOURS SINCE 1969

Thank You IIT Madras

It was in your hostel room that the idea was conceived



The first issue in Jan '69



The Jan '22 Issue

Celebrating 54 years in TECH media

Magazines
- Electronics For You
- Electronics Bazaar
- Open Source For You

Portals
- www.electronicsforu.com
- www.opensourceforu.com
- www.eleb2b.com
- www.electronicsb2b.com

Events
- India Electronics Week
- IOTshow.in
- EFY Expo
- Test & Measurement India
- Open Source India
- Techworldcongress.com

EFY Enterprises Pvt Ltd
D-87/1, Okhla Industrial Area, Phase-1, New Delhi-110020

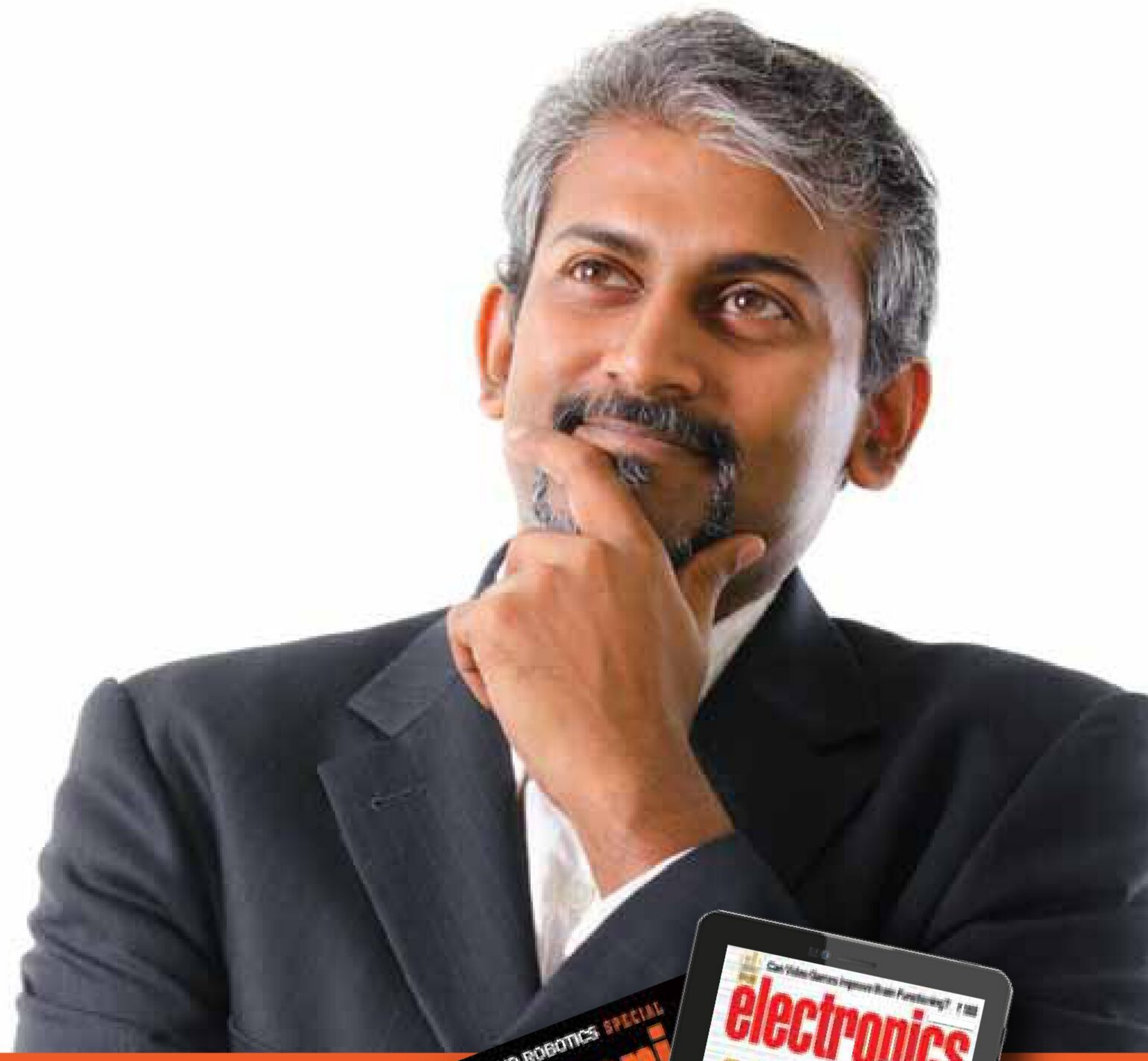
• Phone: +91-98111-55335
• Email: info@efy.in • Website: www.efy.in

R N I No. DELENG/2012/49440, Mailed on 27/28th of Advance month
Published on 27th March 2022
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