



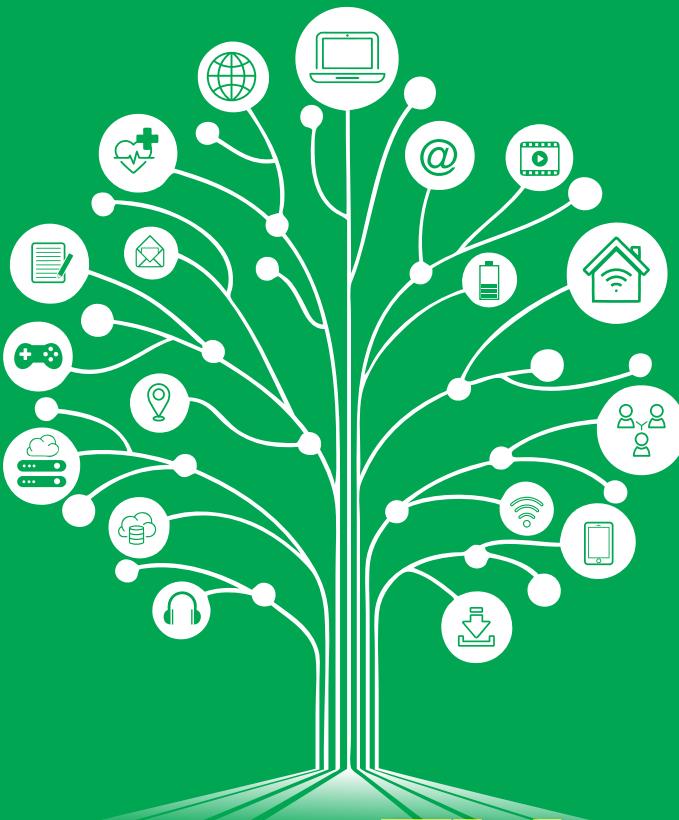
OpenSource

THE COMPLETE MAGAZINE ON OPEN SOURCE

For You

Volume: 08 | Issue: 12 | Pages: 108 | September 2020

An **EFY GROUP** Publication



Edge Computing:
Enhancing
The IoT
Experience

Google Fuchsia:
Improving
On The
Android OS

How The Internet *of* Things Is Evolving

Cloud Based
Implementation Of IoT
Using MQTT Brokers

Open Journey:
Viral B. Shah,
Co-Founder, Julia Computing



10,000+

14,000+
Delegates

340+
Tech Brands

120+
Speakers

10,000+
Sqr Meters

THANK YOU!

Partners, Speakers & Visitors
for making India Electronics Week &
IoTShow.in 2020 a grand success!

See YOU again next year. Mark Your Calendar!

INDIA
ELECTRONICS
WEEK

3-5 FEB 2021 • KTPO Whitefield • Bengaluru

IoTSHOW.IN
PROFIT FROM IoT

Partners of 2020

PLATINUM PARTNER



VISITOR
BAG
PARTNER



BADGE PARTNER



SKILLING
PARTNER



LOGISTIC PARTNER



ASSOCIATE PARTNERS

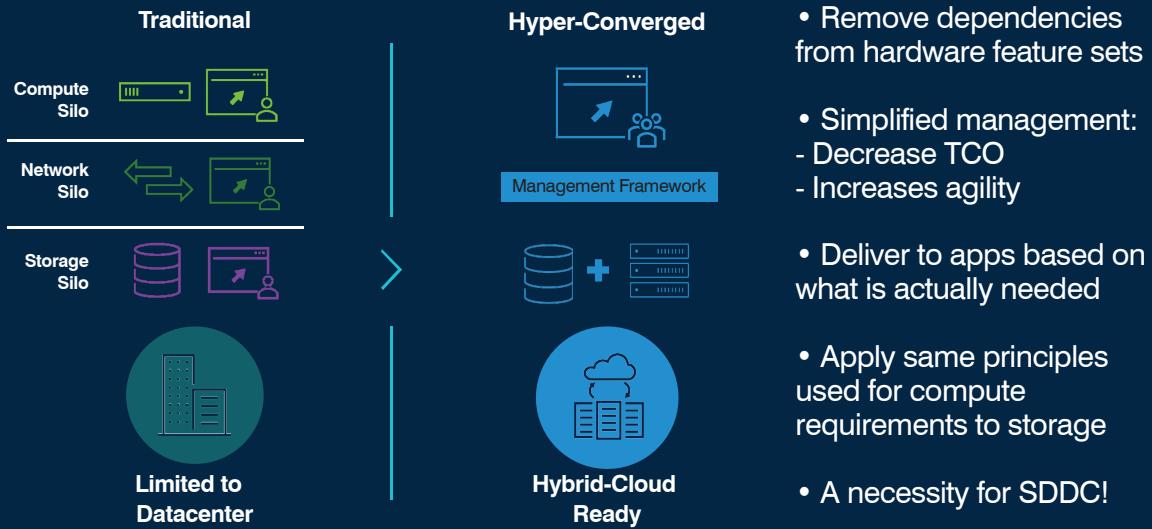


For more details, call: +91-9811155335 or write to us at: growmybiz@efy.in

Do you know how VMWare HCI addresses Infra needs?

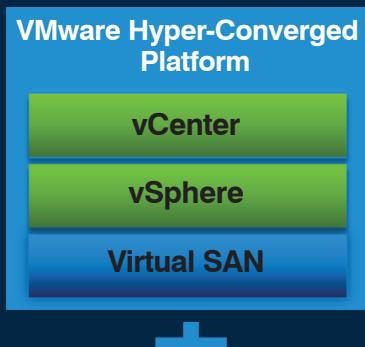
Modern Software-Defined Architecture – VMware HCI

Transitioning from hardware-based solutions to software



Key Tenants of VMware HCI Solution

Choose as per Your Requirement



x86 Intel/ AMD Hardware

VMware HCI Solution Offerings to Choose

VMware HCI Solution [Capacity & Performance]

3 Nodes Solution with

- 60 Cores
- 384GB RAM
- 10TB Hybrid Storage
- vSphere STD
- vSAN STD
- vCenter Foundation
- 3 Years Support

INR 32,62,057/-*

VMware HCI Solution [High Performance]

3 Nodes Solution with

- 60 Cores
- 576GB RAM
- 20TB All Flash Storage
- vSphere STD
- vSAN Adv
- vCenter Foundation
- 3 Years Support

INR 51,90,035/-*

All Prices are exclusive of Taxes. * Installation charges will be extra * No Data Migration included. * This is just an indicative Techno-Commercial offer

* All rights are reserved with IT Solutions India Pvt Ltd to pull this offer back without any notice.

Dipesh Gupta | Email: Dipesh.Gupta@itsipl.com | Call: +91-7290032215

CONTENTS

SEPTEMBER 2020 | ISSN-2456-4885

FOR U & ME

- 22 All You Need to Know About Building a Career in Blockchain Technology
- 30 Bad Users Can Fail A Good AI System
- 36 ApImage: A Promethean Idea

FOCUS

- 40 Cloud Based Implementation of IoT Using MQTT Brokers
- 44 Google Fuchsia: Improving on the Android OS
- 47 IoT, Cloud and Machine Learning: The Building Blocks
- 49 Challenges in Running AI on Embedded Computers
- 51 IoT Creating New Avenues for Data Analytics
- 53 Emerging Trends in the Design of Cognitive IoT
- 55 Internet of Medical Things (IoMT): A Boon for the Healthcare Industry

DEVELOPERS

- 68 Exploring Mainflux and Its Usage
- 71 Managing Project Source Code Development Efficiently with Git
- 75 Use Sourcetrail to Work Easily with Unfamiliar Code
- 78 OpenCV: An Excellent Tool for Computer Vision
- 81 Less is Actually More!
- 83 Let's Get Familiar with Smart Pointers in C++
- 85 Open Source Testing Solutions that Make Software Reliable

COLUMNS

- 88 CodeSport

ADMIN

- 96 Demystifying Containers

OPENGURU

- 103 More Cool FFmpeg Tricks



Overcoming the Evolving DevOps Skills Gap



"With the Internet and open source, the world is your playground" –Viral B. Shah, co-creator, Julia programming



"The adoption of FOSS in the MSME sector needs considerable work"
–Swapneel Patnekar, founder and CEO, Shreshta IT Technologies

REGULAR FEATURES

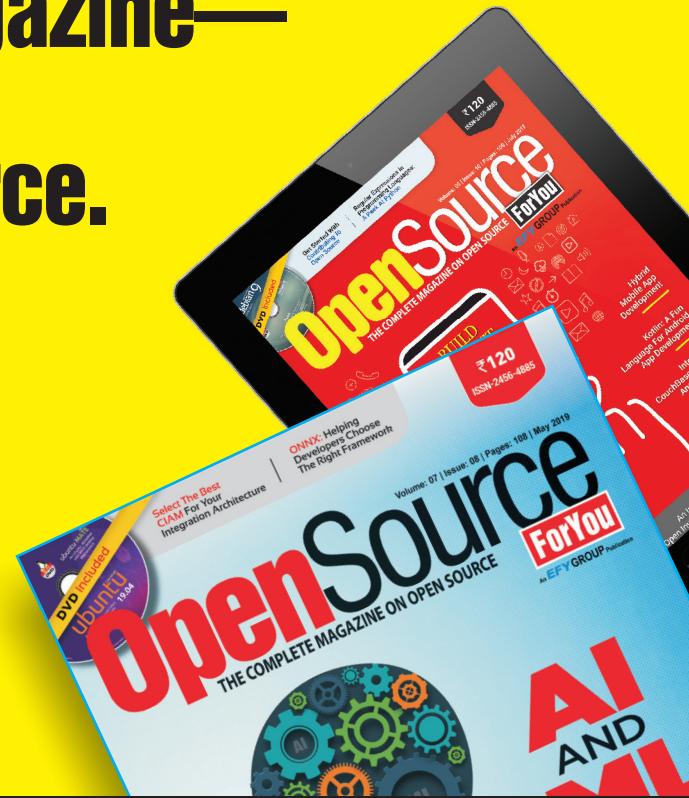
- 07 FossBytes

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

AMAZING OFFER

**Pay for
6 Issues
Get 12!**

(Buy One Get One Free!)



WORLD'S LEADING PUBLICATION ON OPEN SOURCE

Please
Tick (✓)
Your Choice

TEAR OUT

ORDER FORM

TEAR OUT

**Pay for 12 Issues
Get 24 Issues**

and save 50%
(Buy 12 Issues Get 12 Issues Free!)

₹1440

**Pay for 24 Issues
Get 48 Issues**

and save 50%
(Buy 24 Issues Get 24 Issue Free!)

₹2880

**Pay for 36 Issues
Get 72 Issues**

and save 50%
(Buy 36 Issues Get 36 Issue Free!)

₹4320

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

OR

SCAN
THIS
CODE



Name _____ Organisation _____ Mailing Address _____

City _____

Pin Code _____ State _____ Phone No. _____ Email _____

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

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

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

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

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

EDITOR
RAHUL CHOPRA

EDITORIAL, SUBSCRIPTIONS & ADVERTISING

DELHI (HQ)
D-87/1, Okhla Industrial Area, Phase I, New Delhi 110020
Ph: (011) 26810602, 26810603; Fax: 26817563
E-mail: info@efy.in

MISSING ISSUES
E-mail: support@efy.in

BACK ISSUES
Kits 'n' Spares
New Delhi 110020
Ph: (011) 26371661, 26371662
E-mail: info@kitsnspares.com

NEWSSTAND DISTRIBUTION

Ph: 011-40596600
E-mail: efycirc@efy.in

ADVERTISEMENTS
MUMBAI
Ph: (022) 24950047, 24928520
E-mail: elymum@efy.in

BENGALURU
Ph: (080) 25260394, 25260023
E-mail: efylblr@efy.in

PUNE
Ph: 08800295610/ 09870682995
E-mail: efypune@efy.in

GUJARAT
Ph: (079) 61344948
E-mail: efyahd@efy.in

JAPAN
Tindem Inc., Ph: 81-3-3541-4166
E-mail: japan@efy.in

SINGAPORE
Publicitas Singapore Pte Ltd
Ph: +65-6836 2272
E-mail: singapore@efy.in

TAIWAN
JK. Media, Ph: 886-2-87726780 ext. 10
E-mail: taiwan@efy.in

UNITED STATES
E & Tech Media
Ph: +1 860 536 6677
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 © 2020. 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.

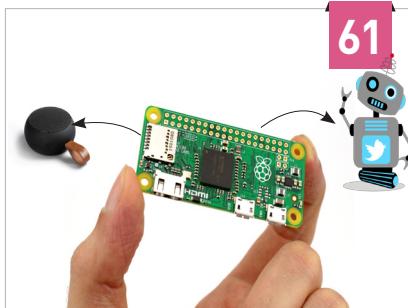
SUBSCRIPTION RATES			
Year	Newstand Price (₹)	You Pay (₹)	Overseas
Five	7200	4320	—
Three	4320	3030	—
One	1440	1150	US\$ 120

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



Edge Computing: Enhancing the IoT Experience



Have Some Fun Doing These Useful Home IoT Projects with Raspberry Pi



Horizontal IoT Platforms: An Overview



Moodle Plugins for Online Education: A Quick Look at Interactive Content - H5P



Emergence of Pen Testing in Cyber Security

New Relic and Grafana Labs team up to advance open instrumentation



New Relic, Inc. and Grafana Labs have announced a partnership to drive advanced open instrumentation and visibility for developers and software teams. The companies said that they have delivered new integrations designed to help engineering teams solve problems even faster.

Under this partnership, Prometheus users can use the Prometheus remote write capability to send metric data directly to New Relic's Telemetry Data Platform with a single configuration change. Grafana open source users can add the Telemetry Data Platform as a Grafana data source using Grafana's native Prometheus data source. This will enable teams to use New Relic's up-to 13 months of retention for their Prometheus metrics, while continuing to use existing Grafana dashboards and alerts.

Raj Dutt, CEO and co-founder, Grafana Labs said, "We are excited to partner with New Relic to expand the number of users who can access Grafana dashboards. We know that organisations have complex technology and vendor ecosystems and our goal at Grafana Labs is to ensure they can get to that elusive 'single pane of glass', no matter where their data is stored. As the creators of Grafana and one of the top contributors to Prometheus, we are excited to formalise our relationship with New Relic and welcome them into the Prometheus and Grafana ecosystems."

Bill Staples, chief product officer, New Relic said, "New Relic is committed to supporting open source software and I am proud to partner with the world's number one open source visualisation leader. Our customers can now visualise their Prometheus metrics stored in New Relic's Telemetry Data Platform using Grafana's world-class dashboards with just one simple config change. This partnership further strengthens New Relic's commitment to advancing open instrumentation and democratising observability for all."

US students survey companies about their adoption of OSS

Tony Wasserman is a professor of software management practice at Carnegie Mellon University Silicon Valley. He is also the executive director of its Center for Open Source Investigation (COSI), focused on the evaluation and adoption of open source software. His team is interested to find out whether there is a correlation between open organisations and their use of open source.

Two of his students have created a survey to gather some data on these topics. The link to undertake this survey is <https://survey.opensourcetrends.com>. It will not take more than 15 minutes to answer the questionnaire, and is totally voluntary and anonymous. One may sign up to receive a report on the results.

Wasserman says, "The goal is to get responses from people who are working in organisations. The first question of the survey is designed to screen out people who don't fit the profile. We believe that both individual developers and their managers are aware of policies, if any, that apply in their organisation about the use of open source software."



The project aims to survey companies about their adoption of open source software, and to see where they are on a scale of maturity, ranging from no use to extensive management and governance of open source use.



Global open source intelligence market to grow at 24.7 per cent from 2020 to 2027

The global open source intelligence market will reach US\$ 29.19 billion by 2027 as per a report by Allied Market Research. The report also said that the valuation stood at US\$ 5.02 billion in 2019, and the market will grow at a CAGR of 24.7 per cent from 2020 to 2027.

As per the report, the rise in the adoption of open source intelligence among multiple organisations, an increase in demand for cloud based open source intelligence among SMEs, and a surge in the need to gain insights for business planning have led to the growth in this market. It also added that a lack of awareness of open source intelligence tools among SMEs and concerns regarding data quality hamper the growth of the market. Developing trends like social media analytics will open new opportunities for market players in the coming years.

The report added, “The outbreak of COVID-19 has greatly affected the global open source intelligence market. During the pandemic, the consumption and growth of data has accelerated, as millions use the Internet to keep in touch with their loved ones, and use online services and entertainment. The governments in several countries have been utilising mobile phone usage and location data to determine population movement patterns and track the spread of COVID-19.”

The report added that the video analytics segment is estimated to show the highest CAGR of 26 per cent during the forecast period. This is due to an increase in demand for enhanced video surveillance in the automotive industry and rise in investment by SMEs in video analytics.

As per the report, by end user type, the law enforcement agencies segment is anticipated to register the fastest CAGR of 28.1 per cent during the forecast period.

Bitdefender releases open source HVI technology through Xen Project

Bitdefender has announced the contribution of Hypervisor Introspection (HVI) to the open source community as a subset of the Xen Project called Hypervisor-based Memory Introspection (HVMI). Bitdefender is open sourcing the mechanisms of HVI used to understand and apply security logic to memory events within running Linux and Windows virtual machines. These mechanisms use virtual machine introspection (VMI) APIs at the hypervisor level.



The code, formerly the intellectual property of Bitdefender, will allow organisations to make sense of the memory view provided by virtual machine introspection within both the Xen and KVM hypervisors, as per the company. HVI takes advantage of the position of hypervisors between underlying hardware and virtualised operating systems like Windows, Linux, desktops, and servers to examine memory, in real-time. It looks for signs of memory based attack techniques that are consistently used to exploit known and unknown vulnerabilities.

Bitdefender is also open sourcing its ‘thin’ hypervisor technology, known as Napoca, which has been used in developing HVI. The company said that Napoca can prove useful to researchers and open source efforts as it virtualises CPU and memory, as opposed to virtualising all hardware. It can also be combined with HVI to protect physical systems.

Shaun Donaldson, director of strategic alliances at Bitdefender, said, “The Xen project is proving extremely fruitful, and the Xen Project hypervisor VMI capabilities have revolutionised security. We are excited to see the range of uses the community will come up with for the technology, and fully expect to see HVI and Napoca technology used in areas beyond the scope of Bitdefender’s security-focused purposes, that we could not anticipate today.”

Offensive Security acquires open source security training project VulnHub

Offensive Security has announced that it has acquired open source security training resource hub VulnHub. It also said that this acquisition is a part of OffSec’s ongoing mission to provide practical training content to aspiring cyber security professionals.

Ning Wang, CEO, Offensive Security, said, “VulnHub’s mission perfectly aligns with our ethos of providing material that allows anyone to ‘try harder’ by gaining practical, hands-on experience in digital security, computer software and network administration.”

VulnHub is an open source, continually updated catalogue of IT assets, which are legally hackable, breakable, and exploitable by design. It added that the



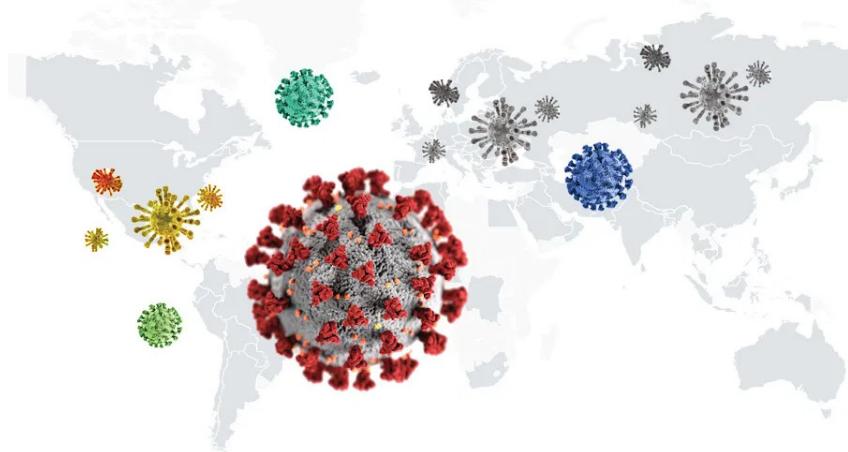
catalogue will help security and IT professionals to try their hand at penetration testing in a safe, controlled environment.

All VulnHub content will remain available for free. OffSec also provides other open source tools designed to help security

professionals gain hands-on experience, including Kali Linux, Exploit DB, and Metasploit.

The Linux Foundation launches Linux Foundation Public Health initiative to combat COVID-19

The Linux Foundation has launched a Linux Foundation Public Health (LFPH) initiative to help public health authorities (PHAs) around the world combat COVID-19 and future epidemics. The initiative is driven by seven members —Cisco, doc.ai, Geometer, IBM, NearForm, Tencent, and VMware — and two hosted exposure notification projects, COVID Shield and COVID Green, which are presently being deployed in Canada, Ireland, and several states in the US.



Dan Kohn, LFPH general manager, said, “To catalyse this open source development, Linux Foundation Public Health is building a global community of leading technology and consulting companies, public health authorities, epidemiologists and other public health specialists, privacy and security experts, and individual developers. While we are excited to launch with two very important open source projects, we think our convening function to enable collaboration to battle this pandemic may be our biggest impact.”

COVID Shield was developed by a volunteer team at Shopify and is in the process of being deployed in Canada. COVID Green was developed by a team at NearForm as part of the Irish government’s response to COVID-19. It has received a high adoption of over one-third of the country’s adults since being deployed by Ireland’s Health Services Executive.

Along with COVID Shield and COVID Green, an earlier cross-industry collaboration effort, the TCN Coalition, is merging into LFPH. This is a global community of technologists supporting the development of privacy-preserving and cross-compatible exposure notification apps during the COVID-19 pandemic.

Updates to Puppet Enterprise announced

Puppet, the industry standard for infrastructure automation, has announced updates to Puppet Enterprise that will make it easier and faster to automate complex infrastructure operations, as per the company. This will provide improved application deployments, fewer errors and disruptions, and greater operational repeatability and efficiency.

The company said that, with this release, customers will be able to use the thousands of open source and Puppet-built modules available on Puppet Forge as building blocks to quickly compose and manage plans through the Puppet Enterprise console. It added that Platform, DevOps and other teams will be able to expand automation into more far-reaching infrastructure use cases. This can be done by mixing and matching imperative tasks with declarative model based automation, with enterprise-grade visibility and control.

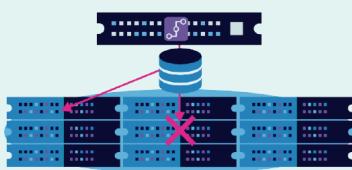
Abby Kearns, Puppet CTO, said, “Application deployments are increasingly waiting on DevOps and platform teams to carry out complex sequences of infrastructure configuration, which, when done manually, also increases the risk of errors and disruption. These teams start off making fast progress with basic automation tools, but hit a roadblock once requirements become more complex. Puppet now provides a growth path from the fast start of Puppet Bolt and Puppet Forge content to powerful orchestration workflows, to deliver on our promise of continuous automation.”

Puppet Enterprise will also include an upgrade pre-check module. It will perform a readiness assessment and identify potential risks, and what changes are needed before running an upgrade.

Commvault upgrades the Hedvig Distributed Storage Platform

Commvault has announced enhancements in the Hedvig Distributed Storage Platform for Kubernetes running in hybrid and multi-cloud environments. It said that the new native API Kubernetes enhancements in the Hedvig Distributed Storage Platform will provide customers the flexibility to develop and run modern applications with encryption and third-party KMIP support for data security in any Kubernetes environment.

Kubernetes has fast become the standard container infrastructure platform for application modernisation. Businesses have to ensure the storage infrastructure supporting containerised workloads has an integrated solution for Kubernetes deployments. The infrastructure must integrate across all types of Kubernetes deployments (cloud-managed or self-managed) and deliver seamless migration, data protection, availability, and disaster recovery for the entirety of these containerised environments.



Phil Goodwin, research director, IDC, said, "Companies are increasingly using Kubernetes for application modernisation. With the new container features for Hedvig, focused on Kubernetes support, Commvault is enabling customers to speed DevOps and apply Commvault's data management capabilities, such as migration and disaster recovery, to containerised environments. These updates to Hedvig make it an intelligent and comprehensive solution for Kubernetes."

SUSE announces enhancements to SUSE Linux Enterprise 15 Service Pack 2

SUSE has revealed enhancements to two enterprise technology solutions — SUSE Linux Enterprise 15 Service Pack 2 and SUSE Manager 4.1. It said that the SUSE Linux Enterprise 15 SP2 helps customers simplify their IT environments by bridging traditional and software-defined infrastructures and breaking through silos of hybrid IT.



The SUSE Linux Enterprise 15 SP2 comes with improved business continuity, as it increases system uptime by up to 12 months with SUSE Linux Enterprise Live Patching. The company said that it also has cloud enhancements with updated cloud images that make it easier for customers to benefit from hyperscalers like Alibaba, AWS, Google, IBM, Microsoft Azure and Oracle. It also provides more security for federal government and public sector customers, and others running high-security environments.

SUSE has also teamed up with Microsoft to develop a solution to collect and graphically display both server and SAP-specific operational data to proactively identify potential problems.

John Gossman, Microsoft distinguished engineer at Microsoft Corp., said, "SUSE and Microsoft have a long-standing commitment to ensuring smooth and reliable implementations of SAP systems. Microsoft Azure has standardised on SUSE's Pacemaker Monitoring Exporter and is pleased to see this capability supported in SUSE Linux Enterprise Server for SAP Applications 15 SP2."

Project ACRN announces v2.0

Project ACRN, an open source IoT hypervisor hosted at the Linux Foundation, has announced ACRN v2.0, which expands the scope of the project. The new version will deliver flexibility in resource sharing, as well as new levels



Stay Connected. Stay Informed. Stay Ahead.

EFY GROUP
YOURS SINCE 1969



SUBSCRIBE AND SAVE

TEAR OUT

ORDER FORM

TEAR OUT

EFY = Electronics For You; EB = Electronics Bazaar

PRINT MAGAZINE(S)	1 YEAR (12 copies each)	3 YEARS (36 copies each)	5 YEARS (60 copies each)
WITHIN INDIA (IN RUPEES)			
EFY (Rs 100/copy)	840 <input type="checkbox"/>	2150 <input type="checkbox"/>	3000 <input type="checkbox"/>
EB (Rs 100/copy)	825 <input type="checkbox"/>	2140 <input type="checkbox"/>	2980 <input type="checkbox"/>
EFY+EB	1585 <input type="checkbox"/>	4075 <input type="checkbox"/>	5680 <input type="checkbox"/>
SAARC COUNTRIES (IN US\$ BY AIR)			
EFY	50 <input type="checkbox"/>	135 <input type="checkbox"/>	NA
EB	45 <input type="checkbox"/>	125 <input type="checkbox"/>	NA
EFY+EB	90 <input type="checkbox"/>	150 <input type="checkbox"/>	NA
OTHER COUNTRIES (IN US\$ BY AIR)			
EFY	100 <input type="checkbox"/>	270 <input type="checkbox"/>	NA
EB	90 <input type="checkbox"/>	250 <input type="checkbox"/>	NA
EFY+EB	180 <input type="checkbox"/>	300 <input type="checkbox"/>	NA

PRINT MAGAZINE SUBSCRIBERS GET:

- Free e-magazine every month
- Free delivery of print magazine by post
- And much more (check: subscribe@efy.in)
- Courier delivery within India @ Rs 50 per copy

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

To subscribe online, visit
<http://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.

NantHealth acquires open source network management application platform OpenNMS

NantHealth, Inc. has announced that it has acquired The OpenNMS Group, Inc., an open source network management company, which will operate as a subsidiary of NantHealth.

Ron Louks, chief operating officer, NantHealth, said, "This acquisition not only expands and diversifies NantHealth's software portfolio and service offerings, but it expands our company-wide capabilities in cloud, SaaS and AI technologies. OpenNMS, we believe, will provide our customers with a new, world class set of services to maintain reliable network connections for critical data flows that enable patient data collaboration and decision making at the point of care."

OpenNMS provides a scalable and comprehensive fault, performance, and traffic monitoring solution that easily integrates with business applications and workflows to monitor and visualise everything in a network. The platform monitors some of the largest networks in the healthcare, technology, finance, government, education, retail and industrial sectors.

David Hustace, CEO of OpenNMS, said, "I'm delighted for our organisation to join the talented team of software and artificial intelligence engineers at NantHealth, which will enable OpenNMS to accelerate its adoption of machine learning and data science technologies. This transaction strengthens our internal capabilities as we move to offer SaaS based solutions to our customers across all industries, while also expanding our penetration in the healthcare industry."

of real-time and functional safety for demanding workloads in both the automotive and industrial segments.

Mike Dolan, senior vice president and general manager of projects at the Linux Foundation, said, "The ACRN project is moving fast to address the increasingly complex requirements for IoT devices, networks and environments. This speed and agility in development can only be achieved through collaboration and we're happy to be able to support this important work."

ACRN 2.0 uses a hybrid-mode architecture to support real-time industrial IoT workloads and edge devices. It simultaneously supports both traditional resource sharing among virtual machines (VMs) and complete VM resource partitioning required for functional safety. It also enables workload management and orchestration, and allows open source orchestrators like OpenStack to manage ACRN VMs. ACRN supports secure container runtimes such as Kata containers orchestrated via Docker or Kubernetes.

Indian government open sources code for Aarogya Setu iOS version

The government has open sourced the code for Aarogya Setu's iOS version. In May, the government had open sourced the code for the app's Android version. It has made the code available on OpenForge, which is the Indian government's own open source platform. The Android version is available on GitHub.

The Indian government has launched the Aarogya Setu mobile app to help limit the spread of COVID-19. This app is available in 12 languages and on the Android, iOS and KaiOS platforms.



Two months back, the government had also launched the Bug Bounty programme that will be hosted by the MyGov team. Under this programme, security experts can get Rs 100,000 on finding security vulnerabilities in the app. There is an additional bounty of Rs 100,000 for code improvement.

The process of supporting open source development is managed by the National Informatics Centre (NIC). Aarogya Setu's source code has been licensed under the Apache License.

The Linux Foundation announces new training course

The Linux Foundation has announced the availability of a new training course, LFS268 – CI/CD with Jenkins X. LFS268, developed in conjunction with the Continuous Delivery Foundation, is designed for site reliability engineers, software developers and architects, as well as DevOps engineers who have to not only master continuous integration (CI) and continuous delivery (CD) but also gain a deeper understanding of the cloud native ecosystem.

Jenkins X is an open source CI/CD tool built on the Kubernetes platform. It takes an opinionated approach to creating CI/CD workflows. This course will introduce the fundamentals of Jenkins X. It will help students learn how to use this tool to create an automated software delivery pipeline. This includes building, testing in various environments, and then promoting your application to production.



The course was developed by Himanshu Gautam, an author, trainer, speaker, and problem solver with over 20 years of experience in software engineering. He is also the founder of Aplima Inc., a CI/CD solution provider.

By the end of the course, students will be able to describe the architecture of Jenkins X, install and operate it, build pipelines in Jenkins X, and create and import existing projects into it. They will also be able to promote applications through various built-in environments, extend and modify pipelines, and create custom build packs.

New earthquake early warning open source project announced

The Linux Foundation has announced that it will host Grillo's OpenEEW project in collaboration with IBM to accelerate the standardisation and deployment of earthquake early warning systems (EEWs) for earthquake preparedness around



the world. This project includes the core components of the Grillo EEW system of integrated capabilities to sense, detect and analyse earthquakes, and alert communities. OpenEEW was created by Grillo with support from IBM, USAID, the Clinton Foundation and Arrow Electronics.

The Linux Foundation launches training course on managing Kubernetes applications with Helm

The Linux Foundation has announced the availability of a new training course, LFS244 – Managing Kubernetes Applications with Helm. LFS244 has been developed in conjunction with the Cloud Native Computing Foundation, and is designed for systems administrators, DevOps engineers, site reliability engineers, software engineers, and others who aim to enhance their operational experience running containerised workloads on the Kubernetes platform.

According to a note released by The Linux Foundation, “Helm is an emerging, open source technology that enables packaging and running applications on Kubernetes in a simple, efficient way. Helm is considered a package manager for Kubernetes, similar to ‘apt’ or ‘yum’ on various Linux distributions, or ‘brew’ on macOS. Using Helm, you can package, share, and install applications built and designed to run on Kubernetes.”

The course provides a deep dive into Helm and how it's used in real-world scenarios to manage the life cycle of applications on Kubernetes. It covers a wide range of topics like the history of the Helm project and its architecture, how to properly install the Helm client, the various components of a Helm chart, and how to create one.

The US\$ 299 course fee provides unlimited access to the course for one year, including all content and labs. Those who complete the entire course will receive an LFS244 verifiable badge, which will demonstrate that they have the ability to successfully deploy and manage container based applications on the Kubernetes platform using Helm.

QuickLogic joins CHIPS Alliance to enhance open source FPGA efforts

QuickLogic Corporation, a developer of ultra-low power multi-core voice-enabled SoCs, embedded FPGA IP, and endpoint AI solutions has joined the CHIPS Alliance. This alliance is a consortium for advancing common and open hardware for interfaces, processors and systems.

Brian Faith, president and CEO at QuickLogic, said, “Over the past few years, the electronics industry has seen a big shift towards open source hardware and software, and we are proud to be one of the companies at the forefront of that movement. We have already been working closely with several CHIPS Alliance members to make FPGA tools and devices more accessible, and we look forward to continuing these efforts as an official member of the organisation.”

QuickLogic recently announced the QuickLogic Open Reconfigurable Computing (QORC) initiative, which aims to broaden access to open FPGA technology for embedded systems developers. QuickLogic’s initial open source development tools, developed in collaboration with CHIPS Alliance members Google and Antmicro, consist of complete support for QuickLogic’s EOS S3 low power voice and sensor processing MCU with an integrated embedded FPGA (eFPGA), and its PolarPro 3E FPGA family.

Dr Zvonimir Bandić, chairman of the CHIPS Alliance, said, “The CHIPS Alliance is continuing to focus on expanding its member base with organisations from a diverse set of industries. QuickLogic, a leader in open source eFPGA IP and FPGA tooling, will help us drive innovation in the FPGA sector and further our mission to remove barriers for open hardware design.”

Earthquakes often have the most severe consequences in developing countries. Nearly three billion people globally live with the threat of an earthquake and don’t have access to nationwide systems. These can cost upwards of US\$ 1 billion. OpenEEW wants to help reduce the costs of EEW systems and accelerate their deployments globally.

The OpenEEW project includes several core IoT components like sensor hardware and firmware that can rapidly detect and transmit ground motion. It comprises real-time detection systems that can be deployed on various platforms, ranging from a Kubernetes cluster to a Raspberry Pi.

Grafana Labs raises US\$ 50 million and announces Grafana Accelerator Program

Grafana Labs has raised US\$ 50 million in a Series B funding round. The fund will be used to aggressively accelerate its product roadmap in key areas across its open and composable observability platform. This will allow customers to mix and match best-of-breed open and proprietary data sources, and nurture and invest in the wider Grafana community, said the company.



The funding round was led by Lightspeed Venture Partners. The oversubscribed round also included additional investment from Lead Edge Capital. Both are existing investors that participated in the company’s first institutional financing in October 2019.

Raj Dutt, co-founder and CEO of Grafana Labs, said, “Grafana Labs’ growth is a testament to one of our core missions of helping people unify and understand their data wherever it lives, giving them complete freedom in choosing their own observability strategy. This allows users to query data natively at its source in real-time, rather than having to move, store and pay for data in yet another database. Their existing data can be combined with Prometheus metrics and Loki logs, and be available on Grafana Cloud, giving them a truly complete understanding of their complex systems.”

Grafana Labs has also announced the Grafana Accelerator Program (GAP) to nurture early stage companies and side projects that are innovating in the wider Grafana ecosystem. GAP will provide free Grafana Cloud and Grafana Enterprise subscriptions, cash grants, equity financing, and inside access to the core Grafana Labs developers.

For more news, visit www.opensourceforu.com



We Influence The Decision Makers!

Unlock Your Potential

With EFY's Business Solutions



“With the Internet and open source, the world is your playground”

Viral B. Shah »



Viral B. Shah is the co-creator of the Julia programming language along with Alan Edelman, Jeff Bezanson, and Stefan Karpinski. The four of them, together with Keno Fischer and Deepak Vinchhi, founded the company Julia Computing. Shah was also involved in the Aadhaar project. He has co-authored the book ‘Rebooting India’ with Nandan Nilekani, co-founder and non-executive chairman of Infosys, that talks about their experience of working on and executing the project. He was awarded the J.H. Wilkinson Prize for Numerical Software for his work on the Julia Programming Language in 2019.

In an exclusive chat with OSFY’s **Sreejani Bhattacharyya**, he talks about his open source journey, growth, challenges, and gives some advice to youngsters starting out.

Shah recalls that he was first introduced to open source software through the *PC Quest* magazine, which used to ship Linux CDs with it. He learnt about Linux, GNU tools like GCC and Emacs, as well as the writings of Richard Stallman from that magazine. He was impressed by what Debian was doing and soon became a Debian developer. After that, he got busy with his PhD and then started the Julia project with his co-creators. Shah has a PhD in computer science from the University of California, Santa Barbara.

He says, “My contributions have always been in the space of scientific

software since I have believed in the power of science and technology to solve some of the world’s most challenging problems. Scientific software plays a special role because it helps us understand, model, design, and predict, making it a powerful tool to solve the challenges we face, whether it is COVID-19, climate change, clean energy, conservation, and even social inequality and justice!”

Shah’s early contributions were as a Debian developer helping maintain the mosix kernel patches. Since then, his major open source projects have been the Julia programming language and Circuitscape, which is modelling software for ecological conservation.

He adds, “Due to my love for scientific software, a lot of the challenges encountered tend to be mathematical and scientific. In my work with Circuitscape, we faced a number of challenges with devising the right mathematical solvers for large landscapes that span entire continents. So we were not just developing open source software, but also developing new methods in the field of landscape ecology.”

Designing the Julia programming language and making it open source

Shah recalls that when the work on Julia was started, they faced the two language problem. They solved it



by designing the Julia programming language and making it open source. He feels that while many like-minded people saw the potential early on, most did not believe that the two language problem—having two different languages for performance and ease of use — could be solved. He believes that he is still fighting that battle for mindshare, but has come a long way!

Shah says, “As projects grow, one has to often also build out organisational capacity for more contributors. These organisations are not well defined in the traditional sense and are challenging to govern and maintain. We think the Julia community has done immense work along these lines as it has grown from a small seed of four creators to thousands of contributors now.”

As of now, Shah continues to work on both Julia and Circuitscape. At Julia Computing, he is actively working with a talented team in the field of differentiable programming, which combines the areas of programming languages, scientific computing and machine learning.

Powering research in vaccines and drugs for diseases

Says Shah, “The work we are doing will make it possible to combine our model based understanding of science with the data-rich world we have entered over the last decade. Today, Julia is powering research in vaccines and drugs for diseases (including COVID-19), in climate science, new battery technology, and so many interesting applications. I feel blessed to be at the centre of all this, working on technologies that truly will impact the world at large.”

During the early days, when they started seeing industry beginning to use Julia, it became clear to the co-founders that in order to make the users successful, they had to set up a company. Shah says, “Six of us came together to found the company —

Alan Edelman, Jeff Bezanson, Stefan Karpinski, Deepak Vinchhi and Keno Fischer—along with me. The company was set up with the clear goal of providing the best technology for Julia users to develop, deploy and scale their Julia programs.

“Towards that, we built three products — the free JuliaPro distribution for individual users, JuliaTeam for enterprise users, and JuliaRun for deploying and scaling data science and AI codes. In addition, we also created JuliaAcademy.com and open sourced all our training materials.”

He says that none of the founders want the company to move away from its roots of innovation and research. Julia Computing collaborates with a number of partners to build cutting-edge domain specific technologies. An example of that is its partnership with Pumas-AI for pharmaceutical modelling and with Carnegie Mellon University for improving battery technology. It also maintains an active relationship with the Julia Lab at MIT for research into Julia, as well as with NumFocus for the open source Julia project. JuliaCon 2020 was the largest JuliaCon ever, with almost 29,000 attendees, and the third largest number of visitors were from India.

Infinite curiosity and an insatiable mind is what is needed

As someone who has been massively successful in this field, Shah has some sound advice to give to youngsters. He says, “Contributing to open source is a great way to build a career and a CV. This is because your software is then out there, and your contributions are visible publicly. In the pre-Internet days, you were limited to your college in finding mentors. With the Internet and open source, the world is your playground. We have seen so many amazing students from India participate through Google Summer of Code, Julia Season of Contributions,

and Major League Hacking. Many of these students have gone on to become long-term contributors in the Julia community.”

He advises picking an area or project that motivates us, where we will be able to work hard and long hours. Skillsets in languages and technologies can be learned with so many amazing online resources. One must have infinite curiosity, an insatiable mind, and the ability to persevere and work on something for years.

Talking about how India is doing and how the open source landscape has shaped up in the country, Shah says, “A friend once said that India is a country of downloaders — we only downloaded open source software but never uploaded any. I think this is changing in a big way. We see a huge number of contributions from India in the Julia ecosystem. Julia is one of the few big open source projects to have a major contribution from India. I also see many students participating in open source conferences, holding workshops, and interacting online.”

Leverage that mass in IT to build an open source movement in India

Shah feels there is a lot of room for improvement, though. Indian colleges and universities can teach the open source philosophy, and integrate their curriculum with open source contributions. “Our IT industry has millions of programmers. At any given time, so many programmers are on the bench, in between projects. Why not leverage that mass to build an open source movement in India, and put India on the map for open source software? A lot can be learnt from how Google Summer of Code is organised, and replicated at a larger scale in India,” concludes Shah. 

 By: Sreejani Bhattacharya

The author is a tech correspondent at OSFY.

“The adoption of FOSS in the MSME sector needs considerable work”



Swapneel Patnekar is the founder and CEO of Belgaum based Shreshta IT Technologies. A FOSS enthusiast, he organises various workshops to encourage and promote the use of FOSS, including the Belgaum Open Source Meetup. He also conducts hands-on workshops on various open source technologies such as Python programming and the art of the command line (Linux), as well as on building projects using the RaspberryPi, at various colleges across Karnataka and Goa. He is a speaker at various open source conferences and meetups.

In a conversation with **Sreejani Bhattacharyya** of **OSFY**, he talks about his open source journey, the challenges he encountered, and the status of the open source ecosystem in India.

◀ **Swapneel Patnekar**,
founder and CEO,
Shreshta IT Technologies





wapneel Patnekar was fortunate enough to get a dial-up Internet connection at home during the late nineties.

It gave him exposure to alternative operating system software, and system software in general. He says that like most people, the gateway to FOSS for him began with the operating system. Red Hat Linux was what he managed to install, and it kind of stuck.

Swapneel adds, “A slow dial-up Internet connection meant having to rely on CDs, which came along with *Chip* and *Digit* magazines. Both the publications had forums, and I was a lurker on both. The other catalyst was FOSS.IN. I have fond memories of a few that I could attend. Today, I use Debian and CentOS, as well as FreeBSD and OpenBSD, with Emacs being the primary operating system.”

The goal has been to promote and implement FOSS

When asked what got him interested in open source, he says, “The answer is: operating system software. I guess the frustration and pain of dealing with the Blue Screen of Death (BSOD) was too much.” The ability to have complete control over the operating system was something that definitely resonated with him. Things like ‘pipes’ as a concept was mind blowing. And it still is, he says.

Swapneel adds, “I had an interest in networking and had been studying for the Cisco CCNA certification (never took the exam but I still configure Cisco gear till this date). So the benefit of using commands to get things done was already known. With GNU/Linux, the ability to configure file servers using Samba, firewalls using iptables, and DNS server using BIND, tilted my interest towards the operating system, and FOSS in general.”

Swapneel has been implementing DNS and DNSSEC using ISC BIND, Unbound, and NSD from NLnet Labs, as well as PowerDNS dnsdist

and recursor for quite some time now. These excellent open source software have been powering core Internet infrastructure such as root servers since the very beginning.

He adds, “We build and secure networks of MSMEs primarily in Tier-2 and Tier-3 cities. We also set up and manage firewalls, storage systems and VPNs using FOSS. For network operators (ISPs), we set up and manage critical Internet infrastructure such as DNS, along with monitoring solutions and managing security, etc. Even in this space, the goal has been to promote and implement FOSS. On the vendor front, the network operator space is dominated by super large players—mostly closed source and expensive options. Take, for example, a route server. One can install OpenBGPD or BIRD, which allows ordinary machines to be used as routers for exchanging routes instead of buying proprietary appliances. This is a viable and economical option. On the threat intelligence front, we are also a partner of Spamhaus Technology in India.”

The thought behind the Belgaum Open Source Meetup group

But it has not just been a rosy journey. There have been roadblocks as well. As Belgaum lacks an IT ecosystem, the most challenging element was not getting an opportunity to participate in or present at meetups. “It’s not always feasible to make a trip to Bengaluru to attend a meetup and then return,” says Swapneel. So he initiated a meetup group in his city, and began visiting colleges frequently for conducting workshops, etc. In the current situation though, the world has become smaller and he has had the opportunity to not only attend but also present at some of the virtual events.

“Belgaum lacks an IT ecosystem. To compensate for this, in my individual capacity, the original thinking was to run regular meetups; so I started the Belgaum Open Source Meetup group and have been running it since. But with

COVID-19, I have been experimenting with doing webinars, which certainly reach a broader audience but have other challenges such as webinar fatigue, etc,” adds Swapneel.

Words of wisdom

Swapneel has some advice for youngsters starting out in this domain, as these are different times in comparison to when he started, when Internet speeds were abysmally low. He says, “Today, there is a massive information overload. So, the simplest advice to everyone is to ‘focus’. The other suggestion is perhaps to think in terms of ‘why’ rather than just ‘how’ to use a specific tool.”

He feels that from a technical point of view, the Indian open source ecosystem has definitely grown, and there is a definite increase in participation in events by the student community, which is great to see.

Adoption of FOSS in the MSME sector is nascent

Swapneel adds, “Having said that, I would like to point out that the adoption of FOSS in enterprises, especially in Tier-2 and Tier-3 cities, is still very nascent. Take an example of an ERP software; organisations typically are largely unaware (this has been my experience in areas that we serve) of excellent options such as ERPNext. In most cases, they reinvent the wheel by tying up with a local software vendor (closed source, mind you) and get permanently locked in. There is a void in this space and it’s merely a matter of outreach. And in most cases that I know, it’s the ERP software that hinders organisations from the shift to GNU/Linux OS.”

He feels that the adoption of FOSS in the MSME sector certainly needs considerable work. One of the ways to bridge that gap is to educate and promote FOSS at the last mile (in the MSME sector) by organising events in Tier-2 and Tier-3 cities. 

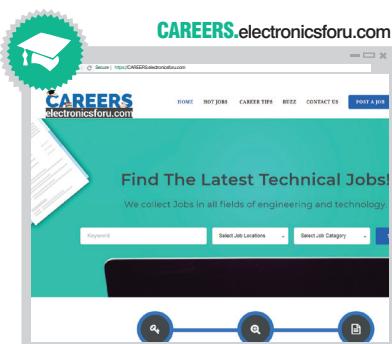
Your Favourite Website Has



The Latest in IOT.

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

A screenshot of the electronicsforu.com website. The header has a search bar and navigation links for DISCOVER, ELECTRONICS PROJECTS, SOFTWARE, and CAREERS. The main content shows a banner for Mouser Electronics and an 'INNOVATIONS' section featuring a project called 'CloudPill: Cloud-based IoT Monitoring of Pillbox Usage'.



Jobs. Career. Technology.

Career advice and jobs related to electronics and IOT.

electronicsforu.com



Design. Electronics. Better.

A resource for professional design engineers.



It's All About Test & Measurement

Great resource for users, buyers & sellers of T&M solutions.

THANKS TO YOU—OUR ONLINE NETWORK IS

FACTS & FIGURES

- Nine 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

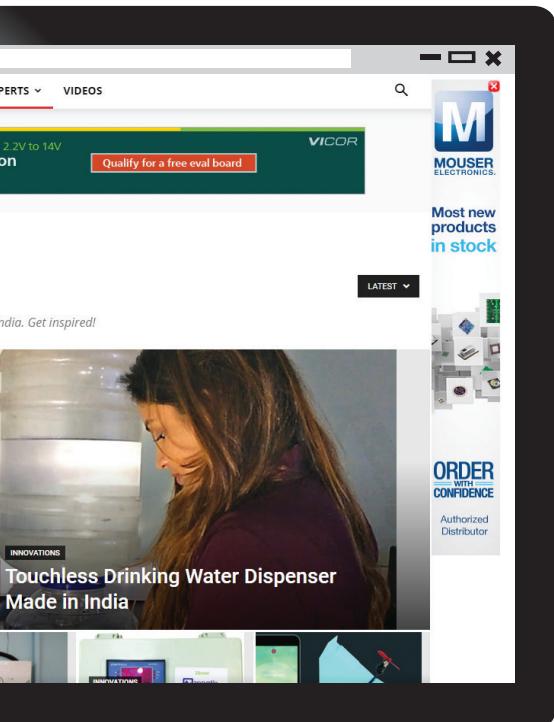
READERS

- 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

Now Grown Into Many!



Amazing DIY Projects. Latest Tech trends.

The hang-out for electronics enthusiasts.



Profit from Technology
Enabling you to benefit from investments in technology.



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



Everything related to Tech & Academia.

A platform for tech faculty and institutions.



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

AMONGST THE WORLD'S TOP 5 AND GROWING!

INDUSTRY

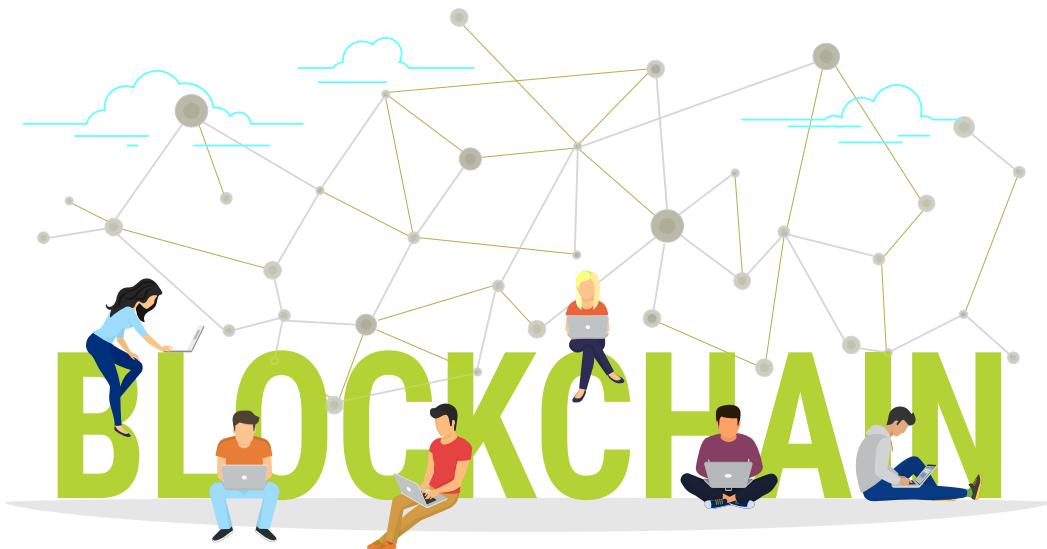
- You can advertise for as little as US\$ 100 per month
- Special combo offers for advertisers in our print publications
- We've now enabled flexible CPM-based advertising
- You can advertise on the platform of your choice (based on your target audience)
- We invite press releases at efy-edit-team@efy.in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

All You Need to Know About Building a Career in Blockchain Technology



The blockchain first came into existence in October 2008, on the foundation of a proposal for Bitcoin, with the objective to create a decentralised, peer-to-peer money network without the involvement of any bank or government. With blockchain technology gaining popularity, there are new job requirements in the field. This article lists the roles, skills, certifications and industries that can be looked at to build a career based on blockchain.



blockchain can be defined as “a time-stamped series of immutable records of data that is managed by a cluster of computers not owned by any single entity.” Every block of data is secured and linked to each other using a cryptographic mechanism, i.e., a chain. In layman terms, blockchain is termed as an encoded, dispersed database that records data similar to a computerised record of exchanges, transactions, contracts, etc. The chief highlight of blockchain is that its automated record is widely available over a large number of PCs,

which means the blockchain system is accessible across a wide network of computers in a distributed fashion. In the real-world, blockchain has already started a revolution in the finance and IT sectors with Bitcoins, smart contracts, Hyperledger and other applications.

Blockchain plays a strong role in building trust across business networks, and it does so using the following five attributes.

Distributed: Blockchain works in a distributed ledger, which is shared and updated with every single transaction happening across the nodes connected to it. Every transaction happens in real-

time without any involvement of the central server.

Secure: The entire blockchain network is cryptographically secured using SHA-256 bit, restricting any sort of unauthorised user access.

Transparent: Every node or participant has the entire copy of blockchain data, so all the transactions can be seen by all the parties connected to the blockchain.

Consensus-based: All the participants must agree that a transaction when it happens is valid, and this is done using consensus algorithms.



Flexible: Smart contracts are executed based on conditions written onto platforms.

In recent years, blockchain has become a trend similar to the way cloud computing was in the mid-2000s and providing professionals, developers and R&D professionals with strong opportunities across various sectors. According to Upwork's skills index, blockchain is the fastest growing skill in the IT sector. As per Burning Glass Technologies, there will be more than 6000+ job openings in this field over the next 5-13 months. So, building a career in blockchain is a good decision in the present scenario. Blockchain technology is transforming companies into more efficient and trustworthy organisations. As it is a new technology, there is a huge shortage of skilled professionals, and big industries and enterprises are leaving no stone unturned to find blockchain experts and giving them a jumpstart in positions and salaries. So, if you are looking for good opportunities in blockchain technology, this article will provide must-needed information with regard to career roles and positions, certifications, technical skills, and even the requirements in the industry.

Career roles and positions in blockchain technology

Like any other technology stack, blockchain needs roles that vary from architects and developers to consultants and testers. However, there are some basic differences as this is an emerging technology where there is good scope for a lot of research related roles. Table 1 lists a few popular roles in blockchain based application development.

There are other intermediary roles depending on the size, volume and complexity of application development, which may also crossover to other disciplinary roles such as AI engineer to implement machine learning based verification

Role	Role description
Technical analyst	Handles technical research with respect to platform suitability.
Platform architect	Overall SPOC for blockchain application design and infrastructure design.
Business analyst/Functional consultant	Helps in understanding the functional and non-functional requirements, and oversees use case development, test case development, test data preparation.
Frontend designer	Prepares the user experience/journey for UI design.
Blockchain engineer (including DevOps role)	Implements the blockchain application and DevOps pipeline activities.
Security consultant	Helps to design the platform security, application security, data security and network/communication security involved in the application design.
Test engineer	Works on test preparation and execution.
Data engineer	Works on data modelling (conceptual, logical and physical), and data driven design related activities.
Design architect	Prepares the solution design along with the platform architect and security consultant for shaping up the components and interfacing activities.

Table 1: Popular roles in blockchain based application development

mechanism in a blockchain platform, Big Data architect who helps to design complex data design in blockchain application design, and cloud architect who helps to design the Infrastructure-as-a-Service design of a cloud based blockchain platform (e.g., Azure, AWS, GCP based blockchain services).

Certifications in blockchain

Some common and popular certifications help you build your career in blockchain technology. The most popular are listed here. There is no recommendation or endorsement of any of these courses. They are merely listed here to show how to walk the path to building a career in blockchain.

University of Nicosia

The University of Nicosia offers multiple blockchain courses and certifications.

- **Blockchain Developer certification:** Intended for developers to understand

blockchain basics, cryptocurrency programming and security frameworks.

- **Blockchain Business Analyst certification:** Intended for consultants to understand regulatory frameworks in cryptocurrencies, and business considerations for design and development of a blockchain platform
- **Blockchain Analyst certification:** Intended for functional and technical analysts to understand open financial systems, digital payment transactions, and application of blockchain platforms in digital payment platforms.

This university is also the first in the world to offer an M.Sc in Digital Currency, which covers blockchain and distributed ledger technology.

Blockchain Institute of Technology (BIT)

BIT offers professional certifications like Certified Blockchain Professional

(CSCP) and Certified Senior Blockchain Professional (CSSCP). CSCP is intended for technical architects who understand blockchain platforms, Bitcoin, business applications of blockchain, cryptocurrency and its usage, development of smart contracts, and design/development of decentralised applications.

CSSCP is advanced professional certification intended to cover the security aspects of the blockchain platform, accounting and smart ledger/distributed ledger management, regulatory and compliance frameworks across different geo/industry standards, advanced smart contract topics, and financial instrument topics.

BIT also offers various online courses on blockchain, Bitcoin, Ethereum and cryptocurrency.

Blockgeeks certification

Blockgeeks offers various short-term courses coupled with certifications on Ethereum platform topics, blockchain for business professionals (functional and non-technical topics), smart contract security aspects, Hyperledger platform and its tools, and the blockchain regulatory framework programme. These certifications help you to earn a blockchain certified developer title in specific topics like Bitcoin, Hyperledger, Ethereum, and regulatory frameworks.

Blockchain Training Alliance (BTA)

BTA offers professional certifications like:

- BTA Certified Blockchain Business Foundation (CBBF) to cover the basics of blockchain fundamentals as well as business topics in blockchain applications.
- BTA Certified Blockchain Solution Architect (CBSA) to cover solution design and development of blockchain platforms and applications.
- BTA Certified Blockchain Developer Ethereum (CBDE) to develop Ethereum platform based blockchain

applications, including programming constructs and practical experience.

- Certified Blockchain Developer Hyperledger Fabric (CBDH) to develop Hyperledger framework and Hyperledger Fabric based applications, including design and development in the platform.
- Certified Blockchain Security Professional (CBSP) to cover advanced security topics like regulatory frameworks, platform security, application security, and data security around various compliance standards.

Exams for these courses are conducted in Pearson VUE centres like for any other professional certification.

Blockchain Council certifications

Blockchain Council offers domain-centric certifications for blockchain professionals like blockchain certified developer, blockchain certified expert, blockchain marketing professional, blockchain HR professional, blockchain certified architect and blockchain finance professional. It also provides specialisation certifications in Corda, Hyperledger, Bitcoin, cryptocurrency and Ethereum (Figure 1).

Technical skills that professionals in blockchain technology need

To take up a blockchain technology based role, it is important to understand various technical aspects and technology frameworks. For instance, one must understand platform security as any failure on this front can cause the entire blockchain based implementation to collapse.

Other blockchain fundamentals that must be understood include basic technical terms such as cryptocurrency, decentralised networks, nodes, mining algorithms, tokens, hash, initial coin offering (ICO), and forks, to name a few. This is important to understand the differences between various blockchain

platforms, and choose the right platform and right architecture for implementing the solution for a given problem.

Technical design of a blockchain platform

A blockchain platform includes the following.

Programming constructs like Java, Python and Ruby, which are required to program the consensus and other decentralised applications in the blockchain platform. Some popular platforms like Hyperledger support a number of programming constructs other than Java and Ruby, such as GoLang, C#, JavaScript, to name a few.

Platform security is fundamental to a blockchain, where one has to understand how to implement an unbreakable architecture to make it reliable for consumers by using various standards like CryptoCurrency Security Standards (CCSS).

Data architecture designs the data model, data flow and data structures to be handled in the transaction to make the entire process efficient, cost-effective, and easy to use for customers.

Application security is designing the security considerations required for various components in the application, including interservice communication and messaging, as well as monitoring and measuring the accessibility to the application.

Security algorithms are the work engines of any blockchain platform, and help to strengthen its computational activities.

Customer Identity Access Management (CIAM) platforms can be any third party integration services like PingIdentity, Akamai, and ForgeRock, to name a few. These services help to transform the customer experience, and handle single sign-on and digital identity for the entire platform.

Machine learning algorithms are a set of algorithms that build a predictive learning model to protect the system from potential attacks.

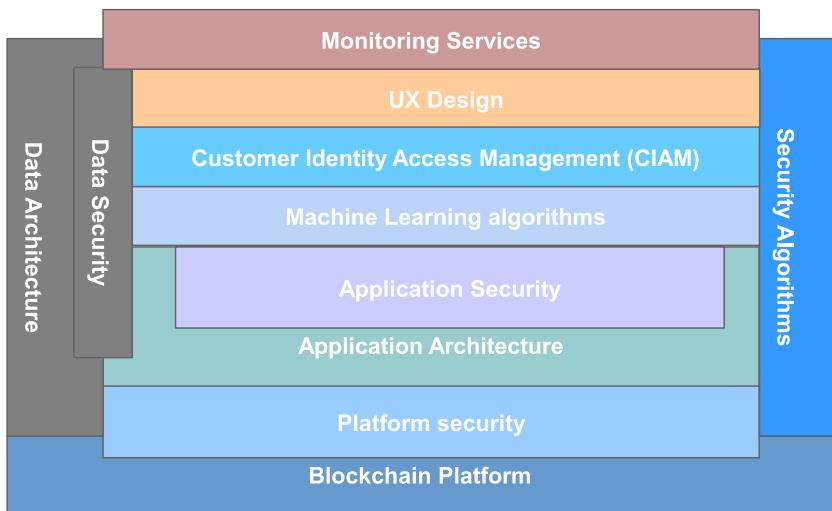


Figure 1: Certifications from Blockchain Council

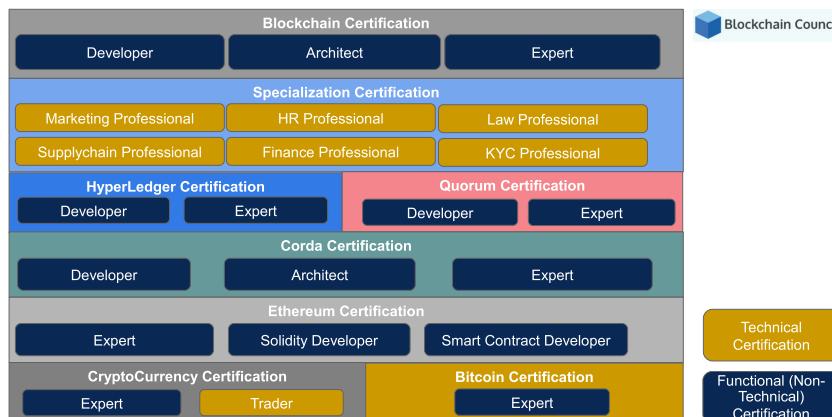


Figure 2: Technical design of a blockchain platform

UX design helps to design a better user experience, making the platform easy to use with a higher return on investment by attracting customers. This is fundamentally a business-engineering concept, used for developing an intuitive and easy to use application design.

Data security and integrity help to handle data models with the highest security features in both data at motion (when it is transformed in any transaction handling) and data at rest (when it is stored in the data store), so that there is no tampering of data.

Monitoring services ensure the entire system is working properly as expected, including alerts and event

management. These also manage health-checks of the entire system in all the layers like infrastructure, applications and data.

The industries in which blockchain will act as a disruptive technology

Here is a list of the sectors blockchain technology promises to disrupt in the near future.

Blockchain in the banking and financial sectors: The banking and finance sectors are using blockchain extensively for transactions using digital cryptocurrencies. As the technology is decentralised, one can use any cryptocurrency variant

to transfer money without the involvement of a third entity. With millions of transactions happening every day in the banking sector, the blockchain has a secure system for maintaining records.

Blockchain in real estate: The real estate industry is plagued by complex procedures and time-consuming paper work. The entire traditional system of documentation can now be replaced with blockchain technology. All documents can be stored safely with less effort and cost, and smart contracts created between buyers and sellers, while funding can be done with consensus algorithms.

Blockchain technology also eliminates third parties by enabling direct communication between two parties.

Blockchain in the gaming industry: Blockchain has the power to unlock amazing benefits for gaming entrepreneurs and developers by preventing fraudulent activities and providing a secure environment for designing, launching and testing games. It even assists game developer companies to buy and sell games. Popular gaming platforms like Microsoft's Xbox and the Sony PlayStation are also using cryptocurrencies in a safe manner. Players who win at various competitions in gaming conferences can be rewarded with Bitcoins. With a blockchain based solution players can buy gaming accessories without going through a bank to process the transactions. Country barriers are non-existent, too.

Blockchain in healthcare:

The healthcare industry has highly sensitive information and blockchain can be used to reduce the risks of data leakage. Blockchain is highly helpful in transforming medical data and storing records in a convenient manner, avoiding the misuse of data. In addition, it offers authorisation and people identification services.

Blockchain in the legal industry:

The legal industry has multi-dimensional and variable data, and the blockchain can play a key role in storing and verifying transactions securely and with transparency. With blockchain, records can be stored, easily verified and authenticated.

Blockchain can revolutionise operations in the industry using smart contracts for IPRs, financial transactions, etc.

Blockchain in transportation:

In the transportation industry, strong document coordination on shared distributed ledgers helps to eliminate useless paper work. With the help of smart contracts, customs clearance at several checkpoints can be done quickly and efficiently, reducing processing time and effort.

Blockchain helps the transport industry to promote faster delivery by helping to resolve disputes for payments. The processing and administration costs involved in transportation are reduced, as blockchain avoids reliance on paper transactions.

Blockchain in government:

Blockchain technologies can help reduce and eliminate bureaucratic red tape and corruption in government agencies. For example, welfare, disability, veterans' and unemployment

benefits could be more easily verified and distributed, eliminating fraud and waste. Smart contracts could ensure that government funds are released only when certain conditions are met. Security, efficiency, and transparency in government functions could be increased across the board.

Blockchain in education: As the power of online and distance learning grows, so does the need for an independent way of verifying students' transcripts and educational records. A blockchain based system could serve almost as a notary for educational records, creating a way for educational institutions to access secure records and transcripts. In fact, it could also help universities and other large

institutions collaborate. No longer would a student have to wait for the course s/he wants offered at Harvard if Oxford is offering it online; her/his grades and records would also be easily and instantly transferable.

As new technologies evolve, it is important to make sure that industries stay updated with them to keep ahead of the competition. Blockchain technology will impact several industries in the years to come. Security of data is something every business is looking for, and there is no safer option than blockchain when it comes to recording and transferring data. In the near future, the demand for blockchain professionals will rise tenfold and create opportunities for a host of new business ventures. 

References

- [1] <https://blockgeeks.com/guides/blockchain-jobs/>
- [2] <https://www.forbes.com/sites/bernardmarr/2018/06/18/the-best-blockchain-jobs-and-careers-available-today/#28527cc27ca0>
- [3] <https://www.entrepreneur.com/article/314921>
- [4] <https://www.educba.com/careers-in-blockchain/>

By: Dr Anand Nayyar and Dr Magesh Kasthuri

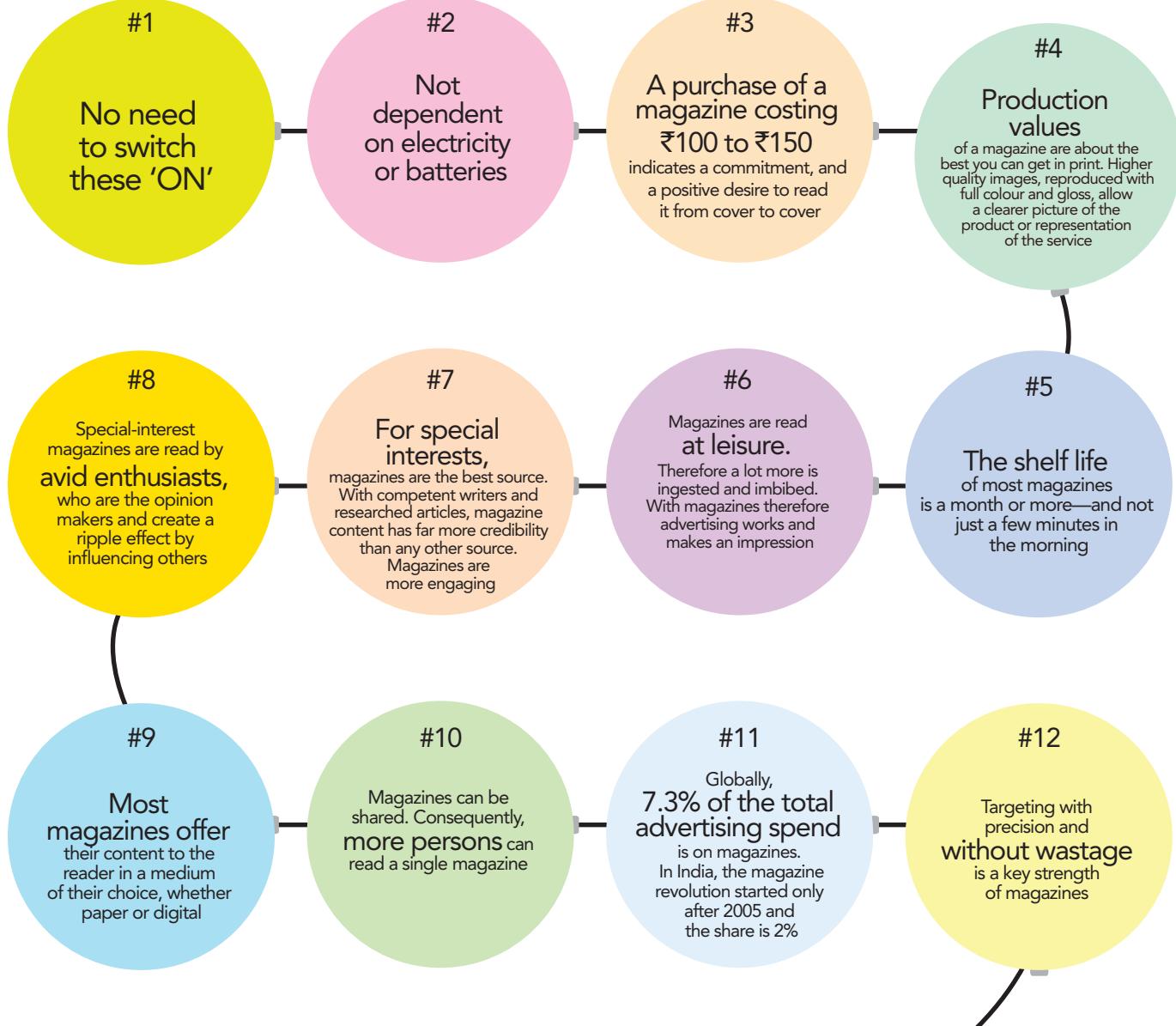
Dr Anand Nayyar is a PhD in wireless sensor networks and swarm intelligence. He works at Duy Tan University, Vietnam. He loves to explore open source technologies, IoT, cloud computing, deep learning and cybersecurity.

Dr Magesh Kasthuri is currently works as a senior distinguished member of the technical staff and Principal Consultant in Wipro Ltd. The article expresses the view of the author and doesn't express the view of his organisation.

OSFY Magazine Attractions During 2020-21

MONTH	THEME
March 2020	Mobile/Web App Development and Optimisation
April 2020	Cybersecurity, Open Source Firewall, Network Security and Monitoring
May 2020	AI, Deep Learning and Machine Learning
June 2020	DevOps Special
July 2020	Blockchain and Open Source
August 2020	Database Management and Optimisation
September 2020	Open Source and IoT
October 2020	Cloud Special: BigData, Hadoop, PaaS, SaaS, IaaS and Cloud
November 2020	Open Source on Windows
December 2020	Open Source Programming (Languages and Tools)
January 2021	Data Security, Storage and Backup
February 2021	Best in the World of Open Source (Tools and Services)

The Power of Magazines!



electronics
FOR YOU
YOURS SINCE 1969

**ELECTRONICS
BAZAAR**

OpenSource
THE COMPLETE MAGAZINE OF OPEN SOURCE

SCAN THIS CODE



To Subscribe, visit:
<http://subscribe.efy.in>

OR

Overcoming the Evolving DevOps Skills Gap

The skills needed for DevOps are changing rapidly.

Companies must recognise this fact, train their teams, and arm them with the tools and techniques they need to enhance their skills.

In 2019, the DevOps Institute's Annual Upskilling Report stated that automation process, cloud analytics and soft skills were the most important. Just one year later, according to the 2020 report, governance, risk and compliance (GRC), process, and knowledge skills increased in importance.

For organisations attempting to bridge the IT skills gap—the areas in which employers have difficulty filling

vacancies—this rapid shift in the importance of skills in just 12 months means that their goal is moving.

The current pandemic has exacerbated this gap. Now, a wider array of skills is necessary for organisations that must shift to remote work and distributed IT environments. For example, a year-over-year uptick in GitHub merge pull requests may signify a heightened need for collaboration or integration skills,



as distributed teams make use of different tools and make more frequent changes to code bases.

Companies that do successfully close the IT skills gap can improve the speed of transformation for both organisations and individuals, and enable more resilience in the face of future disruptions. To close the gap, organisations should understand why certain skills are in demand, select the right tools to help their developers, and instill a culture of collaboration and learning.

Why these skills?

Understanding the cause behind the requirements for these skillsets is the first step to equipping DevOps teams with the approaches, tools, and mindset necessary to bridge the skills gap.

That the need for GRC skills is rising is no surprise. The General Data Protection Regulation (GDPR), California Consumer Protection Act (CCPA), and a myriad of other privacy regulations mean the privacy landscape is always evolving. To ensure organisations can comply with these directives, as well as any future regulations, developers must engrain risk controls and privacy into the software they develop from the ground up. DevOps teams may not need a hard set of skills for governance and compliance but must adhere to privacy-centric development philosophies.

An increasing need for process and knowledge skills may coincide with the rise of site reliability engineering (SRE) practices. SRE teams typically help make life easier for IT. Learning about alert monitoring and building escalation processes are ways individuals can round out their skillsets with an eye on process-related techniques.

By identifying the reasons for changes in skill requirements and understanding which skills may persist, companies will be able to implement more effective upskill-related strategies.

The right tools make everyone's life easier

It's clear that in-demand skills don't always remain in vogue for very long. To help limit the variability in expertise needed from year to year, companies should invest in tools that don't constantly require learning new techniques to operate and that can automate tasks whenever possible.

For example, a growing number of companies work with multiple cloud providers to ensure their applications and services are always available. While a multi-cloud strategy offers benefits, it also means running different projects on different providers' clouds. To limit the amount of skills needed, companies can select container tools that deploy easily to multiple cloud environments without significantly affecting application topology.

Furthermore, tools that automate repetitive processes can help your company reconcile a skills gap. Leveraging solutions that automate processes tied to risk, compliance and governance can help people focus on their core responsibilities and objectives, rather than conducting manual data analyses or attempting to learn data privacy law.

A culture of learning will keep skills fresh

Thoughtfully employing technology can also help close skill gaps. With everyone now working remotely, there are fewer opportunities for in-person training and mentoring. Still, companies must train employees and promote a culture of learning.

Leaders who are proactive in helping employees complete training classes, attend online virtual conferences, and otherwise invest in their skillsets are more likely to have a workforce that remains current on the latest technologies and techniques.

An additional way to promote a culture of learning is by creating events and processes structured around learning, such as:

- Hackathons, a popular way to promote agile cultures; and,
- Communication, which will also remain important as companies move forward in challenging economic environments.

Constantly inspecting processes and analysing their efficacy will help reinforce the skills people are learning, and drive the idea of an agile mindset.

A careful focus on people, processes and technology is crucial to success

Companies will continue to scale up, and paradigms such as multi-cloud and edge computing will continue to experience rapid adoption. Leaders who take the steps now to address the skills gap in a sustainable way will emerge with a more adaptable organisation. But, simply asking developers to learn new skills without understanding why these are necessary or for how long they'll remain crucial is a short-sighted approach. Instead, companies must understand the root causes of changes in skills requirements, and equip their teams with the techniques and tools they need to not only bridge the current skills gap but also to ensure ongoing success in the face of future changes. 

 By: Stephen Chin

The author is the senior director, developer relations at JFrog.

BAD USERS CAN FAIL A GOOD AI SYSTEM

A good artificial intelligence (AI) solution in the hands of bad users can be disastrous, while an average AI solution in the hands of good users can be a great success.

Hence, it's important to educate the users to extract maximum positive value out of it.



If users or other interacting systems are not good enough, then no matter how intelligent your artificial intelligence (AI) system is, it will eventually fail to deliver. The failure may not be the only outcome, but in some cases, it may also result in business risks.

AI systems are not standalone as these often interact with several other systems and humans, too. So, at each interaction point, there is either a chance of failure or degraded performance.

There are a variety of users

We can classify computer users by their roles or expertise levels. In case of role-based classifications, they look like administrators, standard users, or guests. Whereas, skill-based groupings put them in categories such as a dummy, general user, power user, geek, or hacker.



All these categories have user levels that are just good enough to use the computer or any software installed on it. However, if users' expertise is a border-line scenario of being good enough they would soon become bad users of technology. So much so that they can cause a relatively good computer system to come to a halt, including AI.

Additionally, I have also seen that the following user categories are dangerous enough to cause problems.

Creative folks

Creative users are generally skilled enough to use the tool, but they often do so beyond its specified use. Doing that may often render the tool useless or break it.

I remember an interesting incident during my tenure with LG Electronics. One of the products LG manufactured was washing machines. A typical home appliance that a normal user would use for washing clothes.

However, when there were several field failure reports from service centres, especially from the North-West part of India, we were stunned by the creativity of washing machine users.

Restaurant owners in Punjab and nearby regions in India were using these machines for churning *lassi* at a large scale. Churning *lassi* requires more human strength due to its thick texture, especially if you are making it in large commercial quantities.

This is why restaurant owners started using top loader washing machines for making *lassi*. However, this caused operational issues due to unintended and unspecified usage of the appliance and resulted in an influx of a large number of service calls. This kind of creativity looks interesting at face value but certainly causes problems with technology tools.

Another example of such creativity is the use of Microsoft Excel in organisations. How many companies

have you seen where Excel is not only used for tabulation and record-keeping but also being used for small scale automation by running macros?

How many times have you seen people using PowerPoint for making reports instead of creating presentations?

All these are creative uses of tools and may be okay to use once in a while. However, the users are mostly abusing the system and tools, which can cause unintended damages and losses to the organisation. These types of users also expose companies to more substantial risks.

These naughty users are not productive and do not mean any direct harm. They are merely toying with the system and may cause unknown issues, especially with AI systems.

If your AI system has a feedback loop where it gathers data for continuous training and adjustments, this may be an issue as any erroneous or random data can disturb the set process and models.

The users that are deliberately acting bad and trying to sabotage the system could be disgruntled employees.

Sometimes these types of users think that the AI system is no better than them, and they must teach it a lesson. They deliberately make attempts to fail the system at every chance they get.

Mostly, deliberate users do it with some plan. These types of users are difficult to spot in the early stages.

Luddites

A classic example of bad users would be Luddites. These are people who are, in principle, opposed to new technology or ways of working.

Luddites were a secret oath-based organisation of English textile workers in the 19th century, a radical faction that destroyed textile machinery as a form of protest. The group was protesting against the use of machinery in a 'fraudulent and deceitful manner' to get around standard labour practices. Luddites feared that the time spent on

learning the skills of their craft would go to waste as machines would replace their role in the industry.

We often use this term to indicate people who oppose industrialisation, automation, computerisation, or new technologies in general. These users, mostly employees, are threatened and affected due to the implementation of new AI systems. If your change management function is doing any good job, these types would be easy to spot.

Bad user versus incompetent user

Incompetence can mean different things to different people. However, in general, it indicates the inability to do a specified job at a satisfactory level.

If users can use the system without any (human) errors and the way they were required to use it, you can call them competent users. Incompetent users often fail to use the system flawlessly on account of their ability (not system's problems). These users often need considerable help from others to use the system.

Bad users, on the other hand, may be excellent at using the system, but their intent is not a good one.

All incompetent users are inherently bad users of the system; however, bad users may or may not be incompetent. The reason why we need to understand this distinction is—one is curable while the other isn't. You can make incompetent users competent by training them, but no amount of training will help intentional bad users.

Importance of change management

Most of the other system interaction related issues are the result of poor or no change in management during the full term of the project.

While AI has the power to transform the organisations radically, substantial adoption numbers are difficult to achieve without having an effective change management strategy

with **8500+**
registrations, we lived
up to our tag of

Asia's #1 Open Source Conference

17-18 October, 2019
NIMHANS Convention Center, Bengaluru

16th Edition



Thank You

VISITORS, for making the event a huge success

SPEAKERS, for contributing your valuable thoughts

PARTNERS, for your support



OPEN CONNECTIVITY
FOUNDATION™





in place. All the bases should be covered before you begin the implementation and continue it as long as it is necessary.

When you have a complete understanding of how an AI solution will help end-users at all levels in the company, it becomes easy to convey the benefits.

Only quoting the feature list of your new AI solution will not help; you will need to explain what exactly the AI solution is going to do and how it will help everyone in doing their job more effectively.

A safer and less risky approach will be to pick tech-savvy users for the first round of deployments. They will not only provide useful feedback about the AI system you're deploying but can also highlight potential roadblocks for a full rollout. Tech-savvy users can help you determine if the AI solution works as expected for their purposes.

These users then become your advocates within the organisation and help in coaching their peers when needed. They also help in creating significant scale buy-in within teams and potentially reduce the number of bad users down the track, too.

Educating users for better adoption

A proper training plan that uses real-life scenarios and hands-on sessions

is needed—user feedback is welcome and will be acted upon before moving forward. Not doing that means you may disgruntle the right talent.

If you want to ensure smooth transitions and user adoption, start user education early in the process. Moreover, tailor it to each stakeholder group. You should provide them with baseline information and knowledge around AI technology as a whole, and then deeper insights and information on the specific application that you're deploying. It will help in setting their expectations. Every involved member must understand the benefits of the AI solution.

By educational initiatives, you can quickly dispel misconceptions about AI. For some of the stakeholders and users, especially the ones unfamiliar with how AI can help, futuristic technologies can be intimidating. This intimidation begets a defensive response and brings out the lousy user in them in various forms.

With proper education, the benefits of AI can become apparent to your team members and thus foster positive uptake.

If you centre the education on the fact that AI solutions will enhance employees' daily work and make it easier to handle routine tasks, make sure to highlight this aspect. When communicating with your employees, focus on the purpose of the change and

emphasise the positive outcomes it would bring.

Even for executive leaders, it is vital to understand what is happening and knowing the capabilities or limitations of the AI system you're deploying. By investing time in acquiring appropriate education, executives will be able to ask the right questions at the right time. Being more involved is necessary for them.

It is hard to recover from a lack of end-user adoption if you haven't invested enough in user education. So, make sure you have spent an adequate budget in educating users for better AI adoption. Create multiple formats that are readily available for various devices, including offline in-person sessions. When you roll out the training, measure the uptake and types of resources employees use most. It tells you which medium is more effective, and you can leverage it some more.

Going all out on education and training materials can minimise the chances of failure when employees start using the systems.

When you deploy new systems, there is a typical spike in productivity loss, which is generally a result of slow adoption and a long learning period. You can minimise this productivity loss with a proper approach. To ensure successful AI deployment, pair education planning with training.

Moreover, as a rule of thumb, education and training should not end after solution deployments. These must become a periodic activity to ensure that you can sustain all the positive gains.

Checking the performance and gaps

It is reasonable to expect a human user to demonstrate the same performance repeatedly for any given set of scenarios. You would also expect other connected systems to exhibit similar consistent behaviour for you to be able to trust the whole system.

It is essential to check performance for consistency and find any gaps as early as possible in the deployment phase. AI systems usually work on proportionate outcomes, and some variation at the solution level is already accepted. When you couple this inherent variation with the variation of several humans and other systems, it can quickly become unmanageable. Although each variation might have been acceptable independently, when combined, it can be problematic and result in poor overall performance.

That's the reason why performance must be checked for these gaps once you deploy the AI solution. When you use your AI solution, several systems interacting with your AI solution may go haywire. If you didn't plan for systematic changes before the deployment, it could soon become a roadblock.

Performing Gauge R&R (Gauge Repeatability and Reproducibility) tests can reveal several actionable findings. It is a statistical test used to identify variance between multiple operators and can be used to test how various users interact with the same system. You can also use it to check how multiple systems interact with your AI solution.

The outcome of Gauge R&R studies gives you an indication of the causes of variation in the performance. These findings can help in formulating training plans for fixing user performance. These can also help you in formulating system change requirements to make them work seamlessly.

Continuously monitoring the user and system interactions and periodically conducting systematic checks (and tests) can help you in managing incorrect usage of your AI solution.

Handling user testing and feedback

No matter how much content you put into training material, it is not always possible to cover all the questions users may have. It makes it essential to establish easy to use and quickly

accessible communication channel between users and responding team.

If you can make it clear who the contact person is, how long it will take to get a response and how to escalate, if needed, it would help in gaining users' confidence and give them clarity about AI deployments. By doing this, you will only encourage users to come to you when they encounter any issues.

Giving them confidence that their feedback is valuable, and you will always take it on board can go a long way. Moreover, once received, do not just consume the feedback but act on it.

Sincerely checking every feedback and fine-tuning your AI application can help in improving users' experience. It can give them confidence in the deployed AI system. Doing this also reduces the number of bad and incompetent users significantly and thereby reduces your overall risk exposure quickly.

Augmenting HR teams

Until now, HR teams have been carrying out responsibilities to manage the performance of the (human) workforce. This is now changing as machines are becoming smarter, and AI is becoming mainstream. So, how do you plan to handle this new type of workforce, which is fully automatic (AI only) or is augmented by smart machines (humans+AI)?

HR teams will have to manage performance gaps and issues related to system malfunctions as well as retraining requirements of humans and machines. If there is any impact on human performance due to poor-quality AI systems, it will have to be handled differently than how they would handle typical human (only) performance improvement.

Generally speaking, AI systems are smart, but they seriously lack the key characteristic of humans, common sense! With the deployment of digital twins of your human employees, it may become an essential requirement.

Humans in charge of powerful technologies would have to be trained, coached, and managed effectively.

It would be a good idea to take steps towards establishing a new HAIR (Human and AI Resources) team or augment the existing HR team and accommodate these new challenges. The development of appropriate policies and procedures must be core to their initial tasks.

Start looking beyond the technology

No matter how smart the technology or AI in particular is, it cannot apply common sense and human perspective.

Therefore, merely nailing the technical element of AI is not enough; you need to balance it with the human aspect. The understanding of the surrounding environment in which you are using AI is crucial.

Technology teams need to demonstrate cognitive intelligence if they want to be successful. As much as the development and deployment of an AI solution are critical, the user aspect is important too. Without proper use (and users), AI success will surely hang by a thread.

A good AI solution in the hands of bad users can be disastrous, while an average AI solution in the hands of good users can be a great success. The users have the full power to make or break it; your goal should be to enable your users and extract maximum positive value out of it. 

 By: Anand Tamboli

The author is a serial entrepreneur, speaker, award-winning published author and emerging technology thought leader.

The article was originally published in the August 2020 issue of Electronics For You.

AppImage: A Promethean Idea

AppImages are distro-agnostic and can run in all available distros. They need not be installed in the system. Users can carry these with them, and start working in a completely new system. The concept could be game changing for developers. Let's see how it works.

As a novice Linux user, I choose a distro based on its architecture, stability, and the apps it can support.

Among these three, I feel apps are an important factor because most Linux users are dependent on their distro's app store or its repositories for their app needs. This is where Linux falls behind the Windows OS.

Generally, Windows users download apps directly from their respective websites, which provide the most recent version of these apps. Also, apps are downloaded as a single package and users can proceed with the installation right away. But in Linux, the approach is slightly different. The user goes to the distro app store or installs the app via the console by issuing a command. Each distro can have varying versions of the same app. So it is safe to say that there's a lack of uniformity within the distros. To receive the most recent version of the software is not possible via the distro's app store or its repositories, unless your distro is a rolling/semi-rolling release. Sometimes, users may face some issues like "dependencies are not matching" or "dependencies are missing". In this case, users may struggle a lot because they have to find out the matching versions of dependencies, and install those versions in the system to finish the installation process.

But what if you can download a recent version of an app and, without installing it, start using it right away?

AppImage is a kind of app packaging system, which eliminates many obstacles



in the traditional Linux app packaging. In this article, we will briefly look at this system and how it works.

What's wrong with the traditional packaging?

Generally, the developer creates an app, and then releases it to the public. For Linux users, the developer needs to follow certain guidelines that are mandated by the various distros, so that the app can be incorporated into that distro's app store. Since Linux is diverse and developers have to make their apps available to all users, they need to repackage the app for other distros.

Except for the rolling release OSs, many of the Linux OSs are derived from the stable distros. Users are restricted to use the app version that is provided by the original distro, mainly because the OS maintainers focus on the security and stability of the OS.

But, sometimes, users wish to use a recent version of the app. This could be difficult if they rely on the distro's app store, because some distros can

take months to provide updates for the apps. Also, when you install an app via the traditional package manager it will look for the dependencies required and the compatibility of the existing dependencies. And if any discrepancies are found, it will find the required packages and ask you to install those too. But this dependency resolution is specifically made for the particular system that you are working on. So, typically, when you want to install the same app in another system, you will need to repeat the whole process.

This situation gets more complicated if the distro is unable to find the dependencies for that app. In that case, you will have to find the compatible version of the dependencies (sometimes in a tarball, in which case, you will need to build these too) and install them separately into the system.

What is an AppImage?

An AppImage is an app bundle that contains the app in a binary form along with its dependencies in a single file.



This is much like the .exe file for the Windows apps. All you need to do is to download the AppImage file, make it executable, and start running it.

Since AppImages are distro-agnostic, they can run in all available distros. They need not be installed in the system. Users can carry these with them and start working in a completely new system.

AppImages are bundled with all the pieces that are required to run that app. As they do not depend on the system libraries or files, they can run in an isolated environment without disturbing the host system. Any user can run these apps without the native permission restriction of the Linux system.

Also, AppImages are independent universal packaging systems; so users get to use the recent version of the app without having to wait for it to be included in the respective distro packaging.

Working with AppImages

To start working with AppImages, all you need to do is download an AppImage file and make it executable. This can be done via GUI or console. In the GUI way, you need to right-click on the AppImage and go to the ‘Permissions’ tab of the file, check the option ‘Allow this file to run as a program’, close the dialogue box, and double click on the AppImage to run it. The console way is fairly simple. Just issue the following command in your console:

```
$ chmod a+x AppImage_Name.AppImage
```

After the execution of the above command, the AppImage becomes executable, and to run it, issue the following command:

```
$ ./AppImage_Name.AppImage
```

When you double-click on them, some AppImages will ask you to install a desktop file. If you click on ‘Yes’, the respective AppImage will be added to your *Application* menu like an installed app.

Once you’ve decided to uninstall an AppImage, you can just delete the AppImage file from the system. However, if you’ve chosen desktop integration for that file, you may leave some residual files in the system, and the deletion of these files is entirely optional.

Each AppImage is executed by mounting the AppImage file with FUSE (File System in User Space), which enables the user to create a file system at the user level, without touching the host file system at the kernel level.

Sometimes a user may want to take a look under the hood of an AppImage. For that, the AppImage needs to be extracted. To extract its contents, we need to provide the path to the AppImage file and choose the ‘extract’ option from the AppImage *help* menu.

For example, we can take the Museeks AppImage, which is placed in a folder called ‘Test_Table’.

To display the *help* menu of Museeks, the following command should be issued:

```
$ /home/tuxmachine/Test_Table/
museeks-x86_64.AppImage.AppImage
--appimage-help
```

To extract the AppImage, the user needs to issue the following command in the console:

```
$ /home/tuxmachine/Test_Table/
museeks-x86_64.AppImage --appimage-
extract
```

This will extract the contents of the AppImage and place them in a folder called *squashfs-root*.

However, the extracting option is not available for all the AppImages. So if it is not possible via console, you can use the *appimagetool* to extract the contents.

Portable mode

AppImage has another awesome feature called ‘portable mode’. This enables the user to carry the app data along with the app itself. This feature is available in the recent versions of AppImage (built in 2017 or later).

Generally, the portable mode is enabled by creating a directory along with the extension (.home or .config) in the AppImage location. For example, if you want to enable the portable mode in the textosaurus app, then as a first step you need to change your current directory to the AppImage location and

```
AppImage options:
--appimage-extract [<pattern>] Extract content from embedded filesystem image
If pattern is passed, only extract matching files
--appimage-help Print this help
--appimage-mount Mount embedded filesystem image and print
mount point and wait for kill with ctrl-c
--appimage-offset Print byte offset to start of embedded
filesystem image
--appimage-portable-home Create a portable home folder to use as $HOME
--appimage-portable-config Create a portable config folder to use as
$XDG_CONFIG_HOME
--appimage-signature Print digital signature embedded in AppImage
--appimage-updateinfo[rmation] Print update info embedded in AppImage
--appimage-version Print version of AppImageKit

Portable home:
If you would like the application contained inside this AppImage to store its
data alongside this AppImage rather than in your home directory, then you can
place a directory named:
/home/arulmagi/Test_Table/museeks-x86_64.AppImage.home

Or you can invoke this AppImage with the --appimage-portable-home option,
which will create this directory for you. As long as the directory exists
and is neither moved nor renamed, the application contained inside this
AppImage to store its data in this directory rather than in your home
directory
```

Figure 1: AppImage *help* menu

make that AppImage an executable one:

```
$cd Test_Table
$chmod a+x textosaurus-0.9.7-f0908a3-
linux64.AppImage
```

The next step is to create a directory to save the app configuration files:

```
mkdir textosaurus-0.9.7-f0908a3-linux64.
AppImage.home
```

After that, execute the AppImage and make some changes in the defaults (for example, changing the font and its size). Now, close the app.

To ensure that the configuration files are created properly, enter the following command in the console:

```
find textosaurus-0.9.7-f0908a3-linux64.
AppImage.home
```

The above command will display the location of the configuration files. For our example, the output will be like this:

```
textosaurus-0.9.7-f0908a3-linux64.
AppImage.home
textosaurus-0.9.7-f0908a3-linux64.
AppImage.home/.config
textosaurus-0.9.7-f0908a3-linux64.
AppImage.home/.config/Textosaurus
textosaurus-0.9.7-f0908a3-linux64.
AppImage.home/.config/Textosaurus/config.ini
```

Updating AppImages

AppImages use the Zsync2 algorithm for updation. This algorithm downloads the pieces that are changed between your local AppImage and the new AppImage, and could be very useful for the large size AppImages.

To update an AppImage you can either use *AppImageUpdate* or *appimageupdatetool*. The difference between the two is that the former is the GUI version and the latter is the console version. Download any one of these according to your needs, make them executable, and run them with the complete path of your AppImage that

needs to be updated as an argument.

As an example, if you want to check whether a new version of *AppImageUpdate* is available, you can issue the following command in the console:

```
./AppImageUpdate-x86_64.AppImage /
home/tux_machine/Downloads/
AppImageUpdate-x86_64.AppImage
```

It will open up a progress bar to let you know the overall updation process.

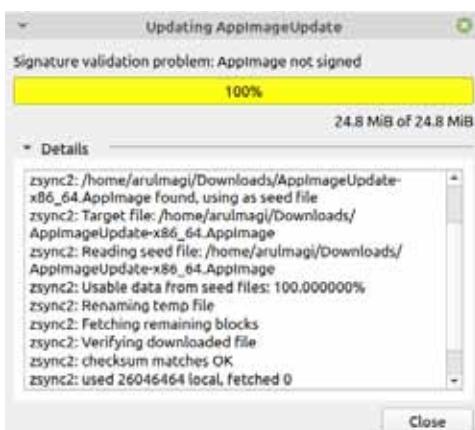


Figure 2: AppImage progress bar

AppImage is fairly young and is evolving, so this article would be incomplete if its pros and cons aren't listed here.

Pros of AppImage

- Even though an AppImage contains the app along with its supporting files and libraries in a single package, its overall size is kept at a reasonable level by employing *SquashFS*, which is used to compress files, inodes and directories.
- AppImages is a decentralised system, so it is not controlled by a single entity. Anyone can craft an AppImage for his or her app, and host it in their site or list it in the crowdsourced directory like *AppImageHub*. Currently, it consists of a total of 689 apps.

3. AppImages are capable of running in an isolated environment; so you can run multiple versions of the same app in a system. This is extremely useful when testing multiple versions of the same app.

4. Since installation is not a requirement for AppImages, they can run in Live OS. This comes in handy when using Linux as a Live OS in a Windows installed system. You can use any AppImage in that environment.

5. AppImages are built on the principle of One App = One File. So maintaining the AppImage is fairly easy, especially for the non-tech users.

6. AppImages are entirely open source; so naturally, no proprietary apps are developed in the AppImage format.

Cons of AppImage

1. AppImages remove maintainers from the equation. Hence, it is up to the developers to ensure that the ingredients of the AppImage should not be outdated and be properly vetted. This is additional baggage for developers.

2. The size of an AppImage is higher than what the traditional *.deb* package may offer. For example, the installation of the *Flameshot* app via the APT method will require packages that are 217KB in size to be downloaded, but the size of the same app in AppImage will be 19.67MB.

3. Apps in the AppImage format are fairly fewer compared to the traditional package managers; however, the number of apps in the AppImage format is gradually increasing.

AppImage is a good replacement for traditional package managers, but it is still a work in progress. Due to its many advantages, it has a lot of potential in the future for developers as well as the Linux community. 

 By: K. Magimai Prakash

The author is a freelance writer, FOSS enthusiast, and a big supporter of decentralisation and privacy, among other things.

FOCUS

Internet of Things - Powered by FOSS

Google Fuchsia: Improving on the Android OS



If hardware is the heart of a computer, then the operating system (OS) is its soul. It's the OS that gives life to a computer and makes it perform various tasks. That's why it's important to choose the right OS for a particular computer system. Fuchsia is an open source compatibility based operating system currently

being developed by Google, which addresses the challenges the Android OS faces. *Read more on page.....44*

Challenges in Running AI on Embedded Computers



This is an extract from a speech presented by Nikhil Bhaskaran, founder, Shunya OS, at iotshow. in 2019. Shunya OS is an operating system that optimises AI libraries for embedded systems, to enable developers to build AI on edge solutions very fast and at a very

low cost. In the next five years, AI is going to be inside everything; and edge analytics is going to be the next big thing for AI. *Read more on page.....49*

Emerging Trends in the Design of Cognitive IoT



This is an extract from a speech presented by Dr Manjunath Iyer, principal consultant, Wipro, at iotshow.in 2019. It describes the emerging trends in the design of cognitive Internet of Things (IoT). Rather than just capturing data,

cognitive IoT devices can think, understand and accordingly provide data to the user. *Read more on page.....53*

Edge Computing: Enhancing the IoT Experience

Edge computing pushes the frontiers of computing, data and services to the extremes of the network, where these are actually needed. It facilitates the collection and analysis of data at its source.

Read more on page.....58

Horizontal IoT Platforms: An Overview

The Internet of Things (IoT) along with Big Data is captivating the digital world. A horizontal IoT platform encourages a large number of clients to work with a typical framework, thus enabling rapid growth and innovation in businesses. By making the gateway and cloud resources have open functionality, innovators can concentrate their efforts on creating devices and services.

Read more on page.....65

Exploring Mainflux and Its Usage

Mainflux is a highly secure, open source and patent-free IoT cloud platform developed in the GO programming language. It is based on a set of microservices. It accepts connections over multiple network protocols like HTTP, MQTT, Websocket, CoAP, etc, making a seamless bridge between them. Mainflux is used as IoT middleware, on top of which advanced IoT solutions can be constructed.

Read more on page.....68

Cloud Based Implementation of IoT Using MQTT Brokers

Message Queuing Telemetry Transport (MQTT) is a standardised, lightweight protocol that transmits messages between connected objects in an IoT network.

A number of devices and gadgets communicate in real-time using advanced wireless technologies. Smart cities, smart offices, smart homes, etc, are interconnected specifically by the IoT in which gadgets and smart objects communicate with each other using specific wireless protocols. IoT is also referred to as Ubiquitous Computing or Pervasive Computing, and there are many real-time applications of these technologies. The FASTags introduced in the country recently to pass through toll plazas is a classical implementation of IoT, in which a small chip on a vehicle communicates with the nearby sensor and a transaction is done automatically.

In such scenarios, the scalability as well as reliability issues are addressed by high-performance network communications or advanced wireless communications using 4G, 5G and related technologies.

Protocols and engines for the implementation of Internet of Things

IoT helps smart gadgets communicate using less resources and minimum power consumption. In the IoT environment, MQTT is a specific protocol for low-powered and lightweight communication so that there is minimum load on devices and gadgets. It is the key data protocol used in IoT for swift communication in low-powered devices.

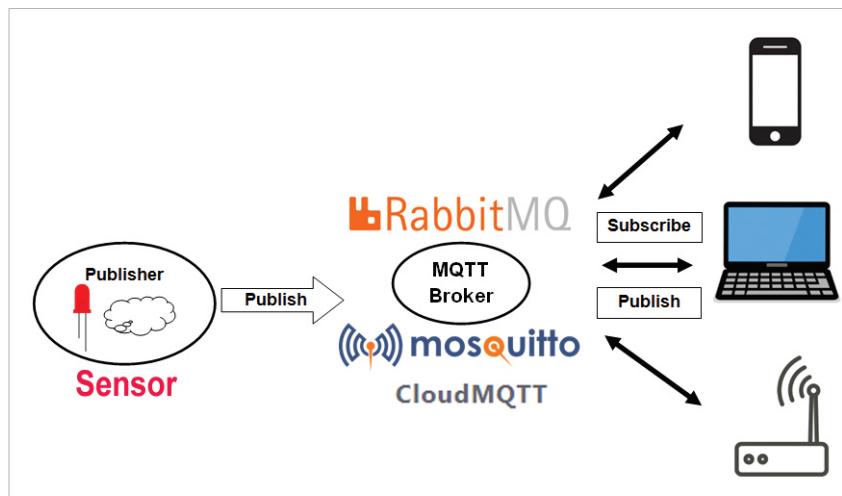


Figure 1: Key constituents of MQTT

Lightweight and low-powered communication needs to be speedy and reliable in order to operate on high frequency network communications with various devices. MQTT enables this communication between a large number of devices and gadgets with varying configurations.

Dimensions of MQTT broker

In IoT communication, there are mainly two types of objects — publisher and subscriber. Data signals transmission is known as publishing. Technically, the transfer of data from the computer to a separate end is called publishing.

Layer	Protocols
Data protocols	MQTT, AMQP, CoAP, Node, Websocket
Infrastructure	6LowPAN, RPL, IPv4/IPv6
Communication transport	Wi-Fi, LPWAN, Bluetooth
Semantics	Web Thing, JSON-LD
Discovery	Physical Web, DNS-SD, mDNS
Device management	OMA-DM, TR-069
Identification	URIs, EPC, IPv6, uCode

Table 1: Key protocols in IoT communication



The sender of data signals is known as publisher and the receiver is referred to as subscriber. The MQTT broker is the central point or simply the IoT server that controls the communication between publisher and subscriber. It handles the data that the publisher transmits for other gadgets known as subscribers, as shown in Figure 1.

For instance, if data is to be transferred from a temperature sensor to a farmer's handheld phone, the data signals are sent by the temperature sensor (publisher) while the farmer's device is the receiver (subscriber) of data.

A number of MQTT brokers are available in the free and open source space including Mosquitto, which implements the MQTT protocol for IoT based communications. It's lightweight, high performance and suitable for use on all devices, from low-power single boards like Arduino and ESP8266 to computers and servers. Instead of a local PC, Mosquitto can be deployed on a cloud based server that implements this broker so that IoT communication is controllable over the Internet.

MQTT broker applications can be installed on specialised devices such as Raspberry Pi, Arduino, laptops, servers, and many others for real world applications.

Cloud platforms for MQTT

A number of MQTT cloud broker services are available, in which the MQTT brokers are available as infrastructure on demand without any physical deployment of these.

CloudMQTT

URL: <https://www.cloudmqtt.com>

CloudMQTT is a commonly used MQTT broker and can be used for IoT interfacing and gadget communication. IoT applications using CloudMQTT include multiple scenarios like smart toll plazas, smart cities, smart homes or smart offices.

MQTT applications	URL
Cloud MQTT	cloudmqtt.com
Mosquitto	mosquitto.org
EMQ X	emqx.io
FLESPI	flespi.com
Bevywise MQTTBroker	bevywise.com/mqtt-broker
M2MQTT	github.com/eclipse/paho.mqtt.m2mqtt
Mosca	github.com/mcollina/mosca
Paho MQTT	eclipse.org/paho
Rabbit MQ	rabbitmq.com
Thingstream	thingstream.io
VerneMQ	vernemq.com
HiveMQ	hivemq.com
MOQUETTE	github.com/moquette-io/moquette
wolfMQTT	wolfssl.com/products/wolfmqtt
EMQTTD	emqttd.io
NET-MQTT	hackage.haskell.org/package/net-mqtt
FLUUX	mqtt.fluux.io

Table 2: Prominent MQTT brokers for IoT based communications

For researchers, the CloudMQTT includes the free package named 'Cute Cat' in which five users/ connections can be created and the IoT devices can be connected, as shown in Figure 2. The free instances can be built on the CloudMQTT network, using the current Google Account.

The key features of Cloud MQTT broker are:

- 24x7 availability of MQTT broker

- Managed Mosquitto servers in the cloud
- Mosquitto implements the MQ Telemetry Transport protocol
- Lightweight methods for messaging using a publish/subscribe message queuing model

The authentication details from CloudMQTT can be used in the IoT hardware or smartphones so that a real-time connection is established

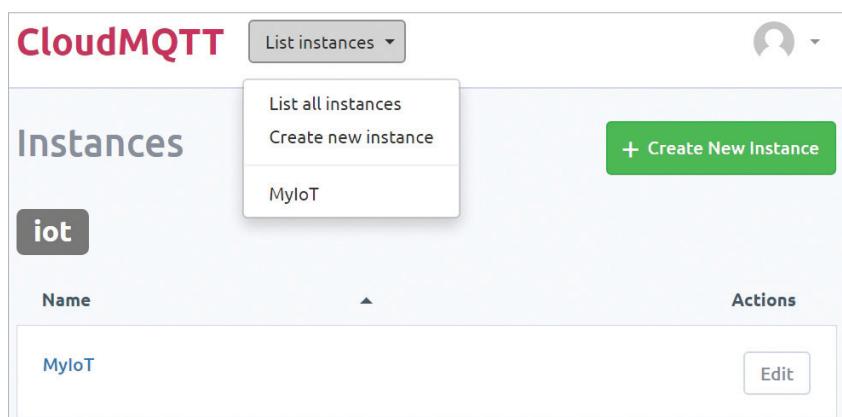


Figure 2: Creating a free instance on CloudMQTT

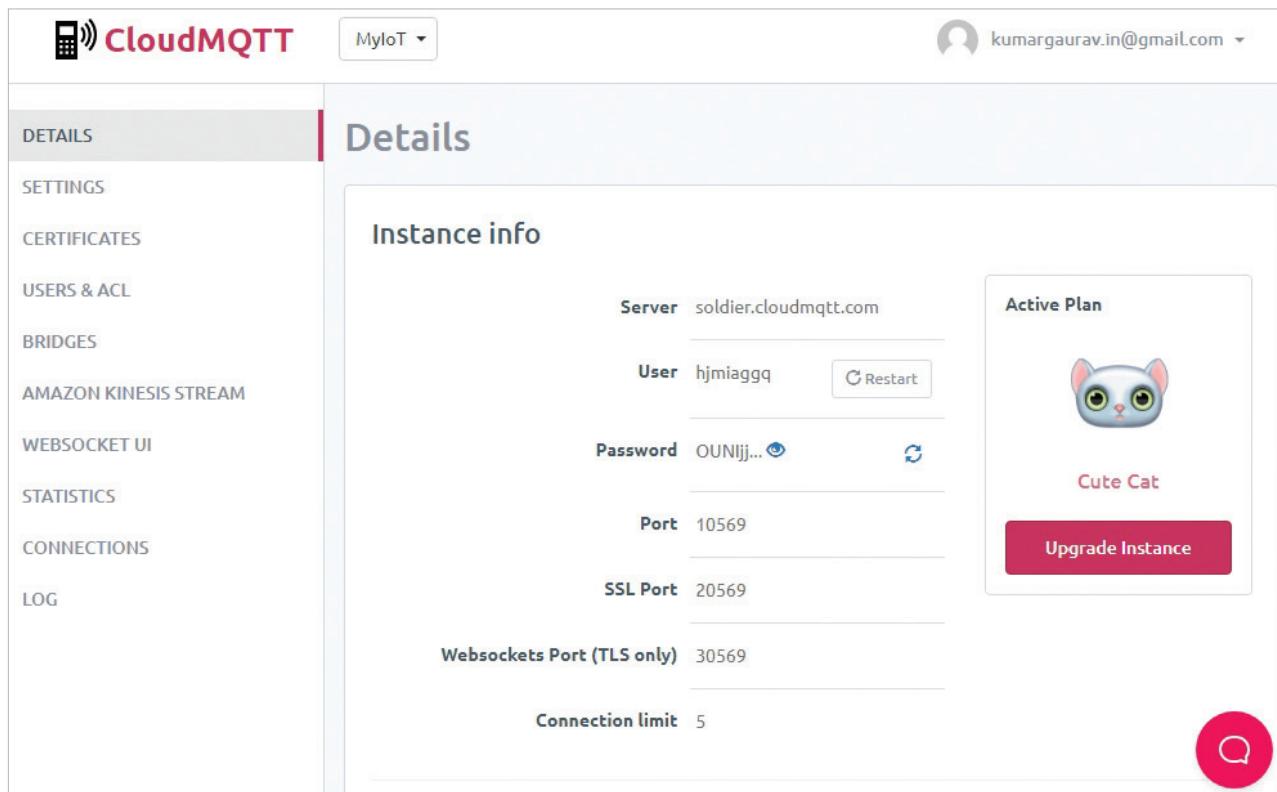


Figure 3: Viewing authentication in CloudMQTT for connections

between CloudMQTT and gadgets, as shown in Figure 3.

A variety of Android applications are available at the IoT MQTT dashboard on Google Play Store, as

shown in Figure 4. Such IoT MQTT applications communicate directly with the MQTT broker platforms so that the IoT apps, tablets and cloud brokers can interact.

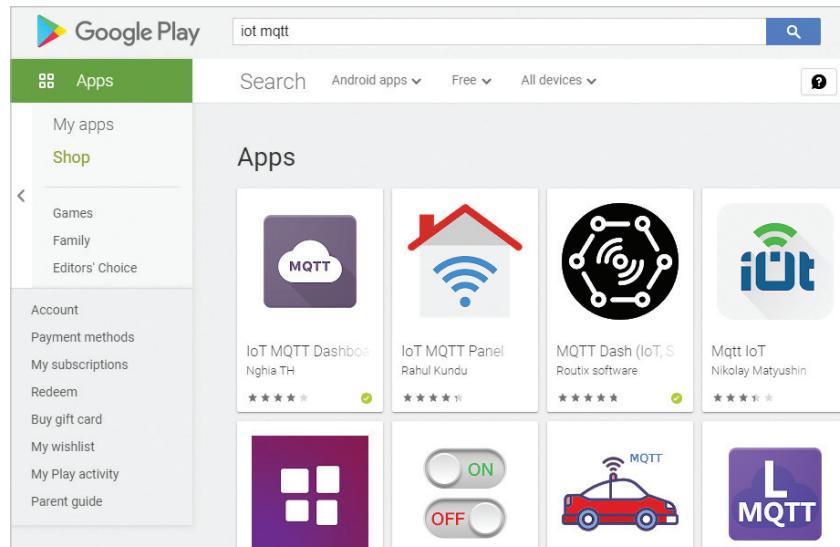


Figure 4: IoT MQTT apps on Google Play Store

DioTY

URL: <http://www.dioty.co>

DioTY is another cloud based MQTT broker application with the free account that can be used to access IoT gadgets, as shown in Figure 5. It provides the scripts so that MQTT broker connections can be rendered to various programming environments.

The following programming platforms can be used for communication between MQTT broker and smart gadgets for publishing as well as subscribing the data and signals:

- C#
- Java
- Python
- Arduino
- Node.JS
- PHP
- Go-Lang
- NodeMCU

Paho Client is one of the

best libraries in Python used for communication and interfacing the devices. For implementation using Python, the package ‘paho-mqtt’ is installed as follows:

```
$ pip install paho-mqtt
```

The following is the code snippet that can be integrated with IoT based devices and gadgets:

```
import paho.mqtt.client as mqtt
# Define event callbacks
def on_connect(client, mySignalData,
CN):
    if CN == 0:
        print("Connected
successfully.")
    else:
        print("Connection failed. CN=
"+str(CN))
def on_publish(client, mySignalData,
MyString):
    print("Message "+str(MyString)+"+
published.")
def on_subscribe(client, mySignalData,
MyString, granted_qos):
    print("Subscribe with MyString
"+str(MyString)+" received.")
def on_message(client, mySignalData,
msg):
    print("Message received on topic
"+msg.topic+" with QoS "+str(msg.qos)+"+
and payload "+msg.payload)
mqttclient = myMQTT.Client()
# Assign event callbacks
mqttclient.on_connect = on_connect
mqttclient.on_publish = on_publish
mqttclient.on_subscribe = on_subscribe
mqttclient.on_message = on_message
# Connect
mqttclient.username_pw_
set('kumargaurav.in@gmail.com',
'7881730a')
mqttclient.connect('myMQTT.dioty.co',
1883)
# Subscribing
mqttclient.subscribe('/kumargaurav.in@
gmail.com/')
# Publishing a message
x=input('Message')
```

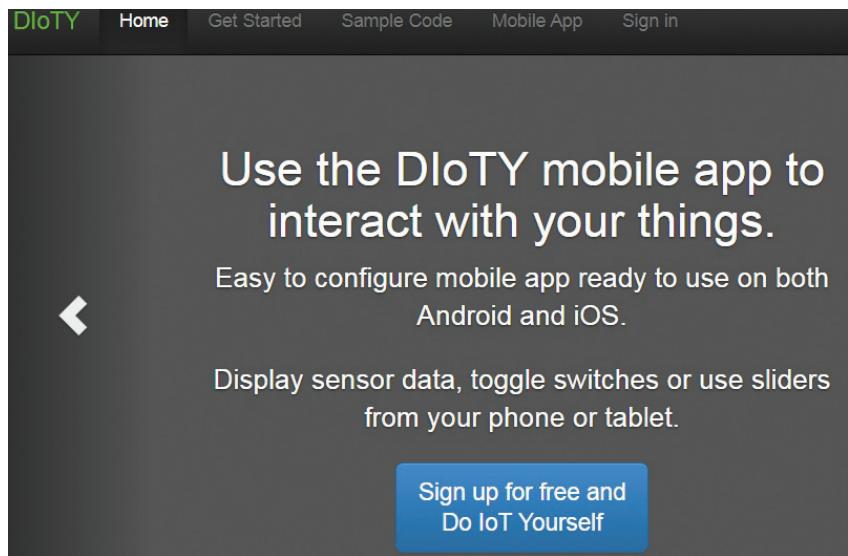


Figure 5: Dashboard of DIoTY

Inject a message here:

Topic: /kumargaurav.in@gmail.com/ Message: Signal Received

Adapt your subscription to view only a subset of your messages:

Topic: /kumargaurav.in@gmail.com/ Keep history

(3:01:13 PM) /kumargaurav.in@gmail.com/: Signal Received

Figure 6: Receiving data on the DIoTY dashboard using MQTT

```
mqttclient.publish('/kumargaurav.in@
gmail.com/', x)
# Loop; exit on error
CN = 0
while CN == 0:
    CN = mqttclient.loop()
    print("CN: " + str(CN))
```

After executing the Python script, the data is handled and processed by MQTT broker, as shown in Figure 6.

Scope for research and development

Research scholars, academicians and practitioners can carry out research and make innovative problem statements on IoT using MQTT brokers. A number of research statements and domains exist in which MQTT based implementations can be done including telemedicine, personal security gadgets, defence equipment, military devices, agribots for smart agriculture, and many others. 

By: Dr Gaurav Kumar

The author is the MD of Magma Research and Consultancy and is associated with various academic institutes, where he delivers expert lectures and conducts technical workshops on the latest technologies and tools. URL: www.gauravkumarindia.com.



If hardware is the heart of a computer, then the operating system (OS) is its soul. It's the OS that gives life to a computer and makes it perform various tasks. That's why it's important to choose the right OS for a particular computer system. Fuchsia is an open source compatibility based operating system currently being developed by Google, which addresses the challenges the Android OS faces.

Google Fuchsia: Improving on the Android OS

The design and development of an OS plays an important role in the performance of computer hardware, and a perfectly designed OS can make the hardware run at its full potential. Basically, there are seven types of OSs – single- and multi-tasking, single- and multi-user, distributed, templated, embedded, real-time, and library. Among these, the single- and multi-tasking OSs are more widely used in many computer systems. The multi-tasking OS, also known as the General Purpose OS (GPOS), is used because it can do multiple tasks at the same

time. GPOSs in use today include Mac, Windows, Linux, Android and Raspberry Pi. They work on a method called time-slicing, whereby the OS slices the CPU time into various time slots. Each time slot is dedicated to a task, which is scheduled accordingly. This process is also known as pre-emptive multi-tasking. The time scheduling of the task is done in such a way that the system throughput is maintained. However, GPOSs don't give time assurance for the completion of any task, because tasks are not executed based on their priority, and can cause latency.

In the case of a Real Time OS (RTOS), every task is executed according to its priority and can be completed within the expected time. That's why RTOSs don't cause any latency like the GPOSs, and are more responsive to the I/O data. As most smartphones use Android and iOS, the user experiences some kind of latency at some point of time because both the OSs use pre-emptive multi-tasking and are not RTOSs. Moreover, Google has developed the Android OS from the kernel of the Linux OS, which was not designed for smartphone hardware. To compensate for this incompatibility,



Android OS uses middleware to support the hardware. Whereas, Apple developed its iOS by shrinking the Mac OS, and as the hardware for the iPhone was designed for iOS, there were no compatibility issues of the kind Android had. The Android OS also suffers from fragmentation at some point of time. For this reason, Android smartphones have performance issues after some use. RTOSs can not only improve the performance of smartphones but also IoT devices. As the I/O response of RTOSs is better than GPOSs, IoT devices can be controlled with minimum latency.

Therefore, to get a better solution, Google is now working on the development of a new OS called Fuchsia. Google Fuchsia is an open source RTOS that is capable of running on many platforms — from embedded systems and smartphones to tablets and personal computers. As Fuchsia is an RTOS, it will not cause any latency and performance issues like the Android OS. In contrast to prior Google-developed operating systems such as Chrome OS and Android, which are based on the Linux kernel, Fuchsia is based on a new kernel called Zircon, named after the mineral. Zircon is derived from Little Kernel, a small operating system intended for embedded systems. ‘Little Kernel’ was developed by Travis Geiselbrecht, a creator of the NewOS kernel used by Haiku. The initial development of Fuchsia has started at GitHub. On July 1, 2019, Google had announced the official website (<https://fuchsia.dev>) of the development project, providing source code and documentation for the operating system.

Requirements of Fuchsia

- Google Fuchsia can be improved and customised using various programming languages and runtimes, including C++, Rust, Flutter, and Web.

- Flutter, a software development kit, is required to code the apps and UI for Fuchsia.
- A 64-bit Intel machine with at least 8GB of RAM and 100GB of free disk space is required to develop Fuchsia.
- One of the best ways to experience Fuchsia is by running it on actual hardware like Acer Switch 12, Intel NUC, and Google Pixelbook.
- The Fuchsia install process, called ‘paving’, requires two connected machines over a network — the machine on which you want to run Fuchsia (‘target’) and the machine on which you build Fuchsia (‘host’).
- Fuchsia creates a UNIX-like file system called MinFS, and can currently support files up to 4GB.
- It requires AArch64 (ARM64) and x86-64 platforms to run.

Functionality and architecture

Google Fuchsia doesn’t use a microkernel but uses a different type of kernel called message passing. A message passing kernel is a kind of Inter Process Communication (IPC) mechanism, where processes communicate with each other by passing messages. Unlike the shared memory kernel, which is also another kind of IPC, a message passing kernel doesn’t share the memory with the processes but uses a message queue instead.

In a message passing kernel, one process can send a message to another through the message queue, and the message carries the information about the requirement. Each message that is sent by a process contains the identity of the process that is going to receive it. Similarly, the process that receives the message creates another message, sends it back to the previous process and informs about the requirement. In this way, messages are passed between every process of the kernel without any flaw.

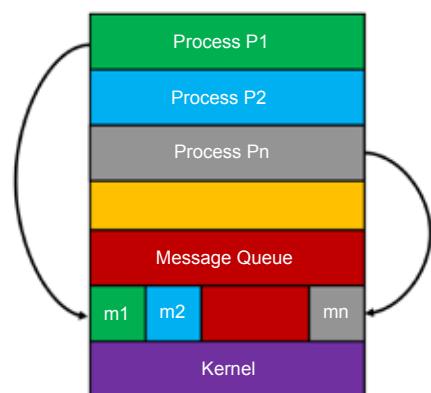


Figure 1: Message passing process

If there are two processes (P1 and P2) running in the kernel, and P1 needs to use a resource that is being currently used by the process P2, then P1 will send a message to P2 to release the resource within a specified time. As soon as P2 receives the message, it will try to complete its process within the time specified by P1, and the kernel will give higher priority to P2. After completing the process, P2 will inform P1 by passing a message that the resource is free to use, and P1 will use the resource after getting the message. This process of message passing can keep the system running without any latency, and is also easier to implement than a shared memory process.

Fuchsia has four layers, as shown in Figure 2.

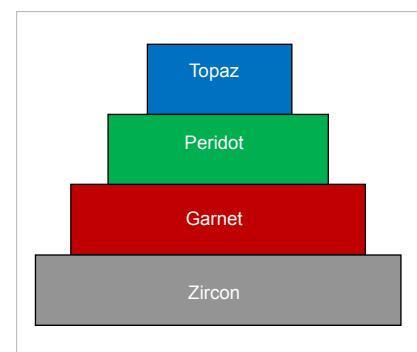


Figure 2: Fuchsia layers

Zircon: This is the first layer and is the kernel of Fuchsia. Zircon is composed of a new kind of message

passing kernel derived from Linux, called the Little Kernel. This is the heart of Fuchsia and has been designed by Google to work with all kinds of devices like smartphones, laptops and IoT devices. This layer mediates hardware access, implements essential software abstractions over shared resources, and provides a platform for low-level software development.

Garnet: This is the second layer and provides device-level system services for software installation, administration, communication with remote systems, and product deployment. This layer contains the network, media and graphics services, and also contains the package management and update system.

Peridot: This third layer provides the services needed to create a cohesive, customisable, multi-device user experience assembled from modules, stories, agents, entities and other components. It contains the device, user, story runners and also the ledger and resolver, as well as the context and suggestion engines.

Topaz: This is the topmost fourth layer that increases system functionality by implementing interfaces defined by underlying layers. It contains four major categories of software: modules, agents, shells and runners. Modules include the calendar, email, and terminal modules; shells include the base shell and the user

shell; agents include the email and chat content providers; and runners include the Web, Dart, and Flutter runners.

Google Fuchsia is coded using the C, C++, Dart, Go, Rust and Python languages. It can support machines with ARM64 and x86-64 architectures. The Flutter development kit produces apps based on Dart that can run at 120 frames per second. Fuchsia also offers a Vulkan based graphics rendering engine called Escher, with specific support for ‘volumetric soft shadows’. A special version of Android Runtime for Fuchsia will be developed, and it will run on machines with this system from a FAR file, which is the equivalent of the Android APK.

Security

Google Fuchsia uses a concept called sandboxing, which prevents the apps from gaining full access to the OS. All access to the kernel of the OS is exposed to the apps as object-capabilities, which means that applications running on Fuchsia have no ambient authority — they can interact only with the objects to which they have been granted access explicitly. Software is delivered in hermetic packages and everything is sandboxed, which means all software that runs on the system, including applications and system components, gains access only to the information it needs to know. Such architecture keeps

the kernel safe, and can prevent any vulnerable attacks.

Features of Fuchsia

- Fuchsia’s user interface and apps are written with a software development kit called Flutter, and it allows cross-platform development abilities for Fuchsia, Android and iOS.
- Due to the cross-platform opportunities offered by the Flutter development kit, users are able to install parts of Fuchsia on Android devices also.
- Fuchsia packages are designed to be updated independently or even delivered ephemerally, and the software is always up-to-date, like a Web page.
- Fuchsia currently supports many types of languages and runtimes, including C++, Rust, Flutter and Web. Developers can use a variety of languages or runtimes without needing to change Fuchsia itself.
- Fuchsia is designed for optimised performance. The use of asynchronous message passing communication reduces latency by letting the sender proceed without waiting for the receiver. It also optimises memory use by avoiding garbage collection in the core operating system, which helps to minimise memory requirements to achieve performance equivalent to other OSs.

Google Fuchsia is a new open source RTOS that will come with various improved features compared to Android, and can help various IoT devices to work effectively. This operating system was chosen by Google to overcome the drawbacks of the Android OS, and that’s why it will give us a better user experience. 

 By: Debojit Acharjee

The author is a software engineer and writer.

Pros and cons of Fuchsia

Pros	Cons
<ul style="list-style-type: none"> • Fuchsia supports many languages like C, C++, Dart, Go, Rust and Python. • It is developed using a development kit called Flutter, which can also be used to create cross-platform apps for Fuchsia, Android and iOS. Fuchsia can also support Android apps (APK) by using a runtime. • It can support both ARM-64 and x86-64 architecture and can run on laptops, smartphones and IoT devices. • Fuchsia uses a sandboxing concept to run the apps, and this makes it a secure OS by preventing any attacks. 	<ul style="list-style-type: none"> • Fuchsia uses a message passing kernel instead of shared memory, the system has slower communication and the connection time between the processes may take some time. • Fuchsia uses a UI that is different from Android, and current Android users might find it difficult to use.

IoT, Cloud and Machine Learning: The Building Blocks

This article touches upon the building blocks that are necessary to enable machine learning in the data received from IoT, and how cloud infrastructure can help if we use the power of open source tools effectively.



MQTT (<http://mqtt.org/>) or Message Queuing Telemetry Transport is an open OASIS and ISO standard (ISO/IEC 20922), lightweight, publish-subscribe network protocol that transports messages between devices. The protocol usually runs over TCP/IP; however, any network protocol that provides ordered, lossless, bi-directional connections can support MQTT. It is designed for connections with remote locations where a ‘small code footprint’ is required or the network bandwidth is limited (Source: <https://en.wikipedia.org/wiki/MQTT>).

Figure 1 explains how to connect the device to the cloud. Let’s discuss the components in detail in the sections below.

MQTT broker

MQTT broker acts as a common point for receiving and publishing messages from clients who are subscribed with it. Clients can connect to the broker, and then receive messages from and also publish messages to the topics. In our case, clients are at the IoT device side and broker resides at the cloud virtual machine. So, the cloud MQTT broker receives data from IoT devices through topics, where devices

publish the messages. Also, the cloud can communicate with devices by publishing messages to topics that are subscribed by the device.

We have a number of options here, and one popular choice is Mosquitto; you can download it from <https://mosquitto.org/download/>.

Eclipse Mosquitto is an open source (EPL/EDL licensed) message broker that implements the MQTT protocol versions 5.0, 3.1.1 and 3.1. It is lightweight, and is suitable for use on devices ranging from low power, single-board computers to full servers. The Mosquitto project also provides a C library for implementing MQTT clients, as well as the very popular `mosquitto_pub` and `mosquitto_sub` command line MQTT clients (Source: <https://mosquitto.org/>).

Eclipse Paho is another option. You can get more details at <https://www.eclipse.org/paho/>.

MQTT client

MQTT client acts as a client connected to the broker. It can receive and publish messages from/to topics that are subscribed by the client from the IoT devices. We have a number of options for MQTT client, and we have already mentioned the Mosquitto C library as MQTT client in the earlier section (<https://github.com/eclipse/mosquitto>). Eclipse Paho client for C is another option (<https://www.eclipse.org/paho/clients/c/>).

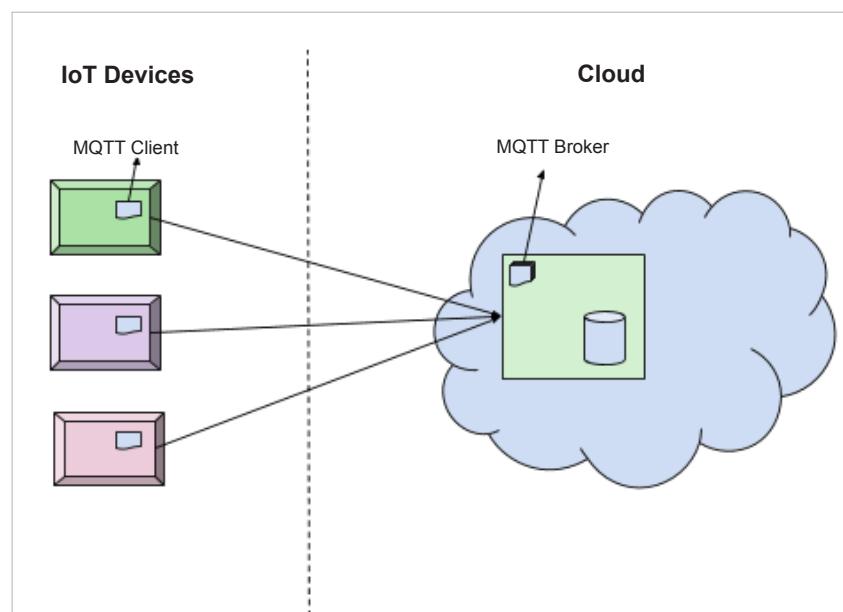


Figure 1: Connecting the device to the cloud



Data transfer from device to cloud

Based on the business use cases, data format and frequency of transfer need to be planned properly. You may have to consider the points given below.

- 1) How does the data help in day-to-day activities, and also for future planning and forecasting?
- 2) How frequently do you need the data? This is based on the data relevance and data changes applicable. For some cases, we may need data from devices each second, and in some cases every hour or day.
- 3) Data from devices may need some transformation while storing it into a cloud database. And based on the business use cases, you may need to display data in your Web/mobile dashboards.
- 4) There may be challenges in the processing of data. For some cases, the system demands sequential data processing (one by one in the way data originated), some cases require parallel processing, and in some cases there may be a pre-condition before the new set of data is processed.
- 5) Processing the data for ML could be another area you may have to plan. Do we need to process hourly, daily, weekly or once a month?

Preparing the ML infrastructure

Since we are choosing the cloud here, we have lots of options, and at the top of the list are AWS, Azure and Google Cloud. All these cloud providers have IoT and ML specific infrastructure and tools, but these are costly and you may not need them in the initial stages. We can create a normal virtual machine (VM) and choose memory, CPU, disk, etc, based on the data and transaction volume.

Given below are the tools and frameworks needed for an Apache Spark based ML infrastructure.

- **Spark 2:** Comes up with all the necessary tools from the Spark ecosystem — Hadoop, Mlib, etc. You can get more details from <https://spark.apache.org/docs/latest/>.
 - **Hadoop:** You can either install Spark2 with Hadoop or Hadoop as standalone.
 - **Python3/Scala/Java:** This depends on what language you prefer to write ML programs
 - **PostgreSQL/MongoDB:** Install this if you have to store data into traditional databases other than Hadoop HDFS for future use and reference.
 - **MLib/Tensorflow/Keras/Scikit learn:** Choose from these ML libraries based on your needs.
 - **Data analytics tools:** Choose these based on your needs.
- The above list is based on the Spark ecosystem; you may have to pick and choose based on the tools and frameworks you are familiar with or are relevant for your business and technology choices.

Common ML use cases in IoT

Given below are a few common use cases based on the data received from devices (strictly based on my experience and may differ in your business case).

1. *Analysis of data patterns for a specific period:* As an example, if data comes from a temperature sensor, then the pattern of temperature data for a location where the device is installed for a day can be analysed for that specific period.
2. *Data missing/changes in duration/changes in pattern, etc:* It is

important to understand the missing data or changes in frequency because immediate action is required, or it can lead to potential errors in our analytics/forecasts.

3. *Inactivities or other ambiguities in data flow:* These errors in data processing must be avoided.
4. *Difference between the forecasted and real data:* This may lead to a correction in data models and trainings.
5. *User and location behaviour from device to device:* Data for each device may differ if user and location behaviour contribute some points to the data.
6. *Frequency of maintenance and root causes for that:* This may be specific to location, usage, transaction volume, etc.

Importance of security

Security is critical if you are handling data. Here are a few things to take care of:

1. Enable SSL/TLS while transferring data from the device to the cloud, to make sure the data is encrypted and secured.
 2. Enable proper security in the cloud to avoid potential data breach or hacking.
 3. Enable the security of Big Data with proper user and group roles; secure the data based on the customers and clients.
- This article is an attempt to give you an understanding of the integration between IoT devices and the cloud. Apache Kafka is an alternative for MQTT, but the advantage of MQTT is its lightweight, hassle-free architecture. 

By: Somanath Balakrishnan

The author has 16+ years of experience in the IT industry and is currently working as a software architect at Digital Core Technologies, Kochi. He blogs at <http://somanath.tv/>.

Challenges in Running AI on Embedded Computers



This is an extract from a speech presented by Nikhil Bhaskaran, founder, Shunya OS, at [iotshow.in](#) 2019.

Shunya OS is an operating system that optimises AI libraries for embedded systems, to enable developers to build AI on edge solutions very fast and at a very low cost. In the next five years, AI is going to be inside everything; and edge analytics is going to be the next big thing for AI.

The work going on in the artificial intelligence (AI) space nowadays is more on the application side to solve industry problems. And people want to get the solution into the market as fast as possible. For this, most of the AI libraries are built into the cloud. In that, there is a large stack, where the first challenge starts with data.

You need to train a model. It is not creating a model that is the biggest challenge, but finding data and arranging it in a way to efficiently train your model is. So, there is a large market for getting the right data.

The second part is building a model. There are many applications that do not require creating a model like face

recognition and object detection. Because these common applications already have ready models available out there—one just needs to choose the right one for the application. And after that, you need to further train your model, as most models are usually trained for different sets of data which do not give a good output.

As each and every library is good at one thing or the other, one needs to carefully choose the right one that is suitable for the application.

After having a working model, it must be used purposefully to generate enough money. Some of the questions which the entrepreneurs should focus on are: What should it (the model) be used for? What is the market looking for?



Apart from all this, there is a major competition since everyone in AI is working directly on the cloud with respect to model training and data. However, AI can run equally well on embedded computers. This is commonly unknown to people.

All AI will happen on the edge

When running your model on the cloud, usually Google engine API or Amazon's object recognition models are used, for which certain amount is paid. Once the solution is built, over a period of time you have to pay a lot of money.

Instead, the same model can be coded onto hardware. On the embedded system you can get the performance of the cloud without having to pay on a long term basis. All analysis happens on the embedded device and it sends only limited data to the cloud.

Edge analytics is going to be the next big thing for AI. All of the data which the AI library is sending down will have computation done by the processor and give results. As an example, in the past, when gaming came in, all the mathematical computation was done in GPU. Now, there are vector processing units on the chips which can process vectors (data) coming from the cloud very fast.

The GPU is also good at doing floating-point calculations.

People are unaware that a lot of AI applications can be built on a chip which the companies can optimise.

The number of libraries available for AI today stands at approximately 800 plus. And this is also the amount of competition in the AI space. This might sound unbelievable, but when computers came in, people never imagined them to turn out to be this big. Gradually, they became a necessity in our lives. And AI is going to be much bigger than that. In the next five years, AI is going to be inside everything. A lot of products will come with AI built into them.

Currently, there are twenty plus companies dealing with AI technology. Some of the popular ones are Tencent, TensorFlow, Caffe, Chainer, ONNX and PyTorch.

Challenges faced in embedded

Challenges such as model size, choosing the appropriate model and framework are commonly faced by people who work on AI. The system side challenges are something that most people are not aware of. These include:

Platform: In the cloud, the code is pre-installed. But for embedded, you need to fetch the source and compile it into the machine.

Fetching: Finding the right source is also a challenge. One needs to do a lot of research to find the right source, the correct patches and configure them.

Architecture support: Mostly ARM is used, which can be either ARM7 or ARM8. With the help of Neon support in ARM8, hardware FPUs (floating point units) provide 5x performance. It is very advantageous but challenging as well.

Compilation: Cross-compiling does not give the required performance. So, to derive performance and make a library work optimally on hardware, you need to do a native compilation. This requires a very long duration of coding and consumes a lot of time.

Installation: You may encounter several error messages after compilation. Care needs to be taken to install libraries correctly, especially for new machines.

Not just these, all libraries usually have some dependencies. Dependencies are some additional packages that are needed to run a library. Each tool/framework/library on an average has 4-5 dependencies.

Those packages also need to be compiled to get optimal performance natively from the AI libraries installed on your embedded system. Software called Docker is the quickest way to run packages, but it is not optimised while it

is running and must be used with care.

There is this perception in the industry that to run AI on the edge, you need a lot of computation or one needs to have better computing power so that enough performance is obtained. However, this is not correct. When that solution is sold, your cost is going to be higher. So, the focus should be on better engineering, which makes the price come down. This should happen without compromising on the latest technological advances.

OpenGL and OpenCL libraries

In an embedded system, while a CPU will be obviously present, GPU and NPU are optional. If you have GPU, then it will have OpenGL or OpenCL library. OpenGL is a graphics library which passes on to GPU the graphic calculations it receives. OpenCL is a compute library which checks the amount of computational power on the hardware; whatever computational request it gets is distributed among the system. By adding OpenCL, your performance goes very high. It is an additional layer which handles requests much better than OS.

There is still a lot to be done on the application side. After writing the code in the cloud, when you run it, the analysis process begins. In this process, the code generates an unknown number of threats which can cause hardware failure if the system cannot handle them. But in embedded, it is custom.

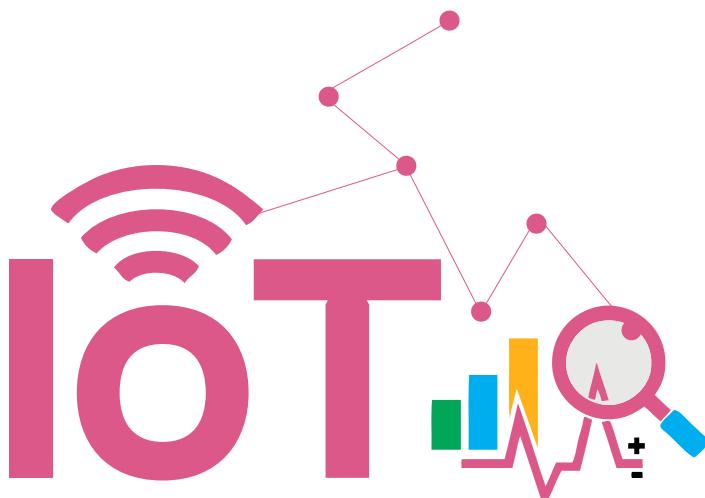
Here, edge computing plays a major role by taking computing actions directly on the hardware instead of sending data to the cloud for computation.

As embedded AI space is coming up, there are some frameworks which are getting built only for this side. Currently, TVM is doing the best in this area. 

By: Nikhil Bhaskaran

The author has more than a decade of experience in core electronics from design to production. He has lived in Shenzhen, China for eight years and has deep experience of embedded hardware. He is also the founder of IoTiot.in—the largest IoT and AI innovators community in Pune.

The article was originally published in the June 2020 issue of Electronics For You.



Creating New Avenues for Data Analytics

An IoT solution is incomplete without in-built data analytics as a part of the user application layers. Customers look for an effortless way to know the insights from data collected by the IoT hardware using sensors. The greatest value lies in how businesses apply IoT, learn from the data captured, and improve efficiencies.

Ihe Internet of Things (IoT) is a network of ‘things’ with in-built intelligence to sense, store and exchange information with mobile phones, machines, smart homes, vehicles, electronic devices, etc. It is a part of the fourth industrial revolution, which is basically an evolution of mobile and embedded technologies

with connectivity to the Internet, integrating greater computing and analytic capabilities. International Data Corporation (IDC) says, “There will be 41.6 billion connected devices, generating 79.4 zettabytes (ZB) of data by 2025.”

With billions of devices connected to the Internet and so many people exchanging information with devices,

these can become an intelligent network of systems. These systems, when sharing data over the cloud, can transform products, businesses, and lives in countless ways—whether it’s fast medical service, better education, lower transportation costs, or optimising energy generation and consumption.

IoT technology stack

The IoT technology stack contains five different layers, including hardware, device software, communication, cloud, and user applications. The whole stack is required to work in sync to benefit from the IoT technology.

The hardware layer contains the microcontroller, necessary sensors, and peripherals suitable for the product. The device software layer guides the hardware to function as per specifications. It is basically a firmware enabling the hardware. The communication layer implements Internet connectivity using wireless protocols like Bluetooth, Wi-Fi, or GSM based on the range, bandwidth,

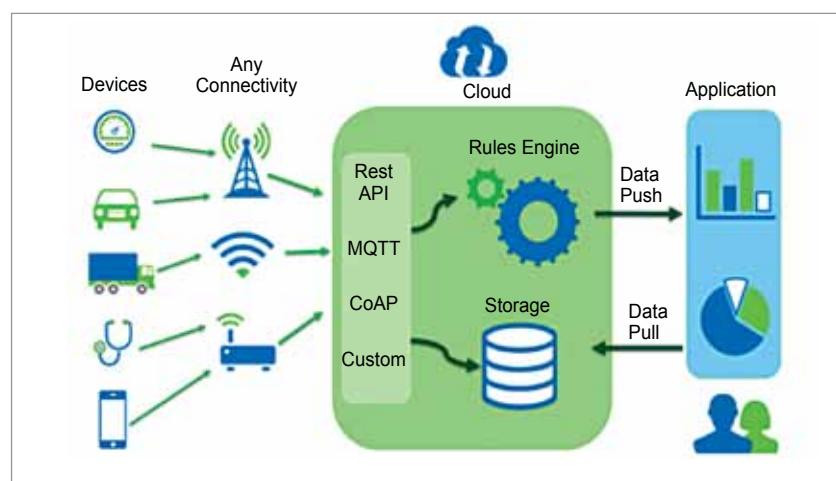


Figure 1: Data analytics application analyses complex data sets generated by different IoT devices and stored in a cloud over Wi-Fi, GSM and Bluetooth connectivity (Credit: <https://itviconsultants.com>)



and connectivity requirements of the device.

The cloud provides additional storage along with local onboard storage to achieve anytime access, data security, and unlimited extendible storage. The user application layer is key to the success of the product as customers get a feel using it.

A user can be a chief technology officer (CTO), analysing a dashboard to understand average defect rate, machine breakdowns, and production numbers, or a software engineer at the office checking what's the average temperature at home to switch on the AC or geyser using a mobile app.

Data analytics

After IoT data analysis, businesses can develop machine learning (ML) models and implement them as part of the IoT solution infrastructure, so that action is taken without any human intervention when a similar pattern arises in the data generated in real time. ML trains machines on how to respond to a combination of inputs, even unknown inputs, and ensure a steady and accurate output. It involves performing data modelling on the data collected by IoT sensors, followed by creating an ML model using statistical or mathematical algorithms. These models are deployed and used until they ensure high accuracy.

In healthcare, the IoT revolution is driving new equipment, remote monitoring, and wearables to constantly monitor the health of individuals. Performing data analytics can help draw useful insights from the data like heart rate pattern, SO₂ oxygen pattern, and average body temperatures. Based on these vitals, doctors can predict symptoms of disease well in advance.

"You can have data without information, but you cannot have information without data," says Daniel Keys Moran, an American fiction writer and computer programmer. IoT is key to look deep into processes and generate related data, whereas

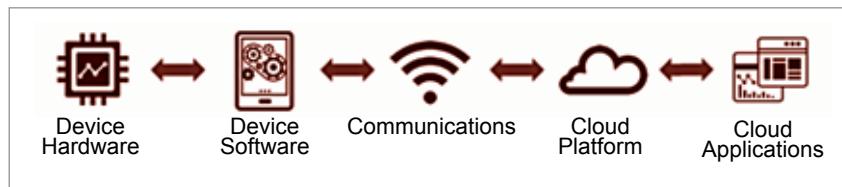


Figure 2: The IoT technology stack contains five layers to provide all the capabilities of an IoT solution (Credit: <https://iotforall.com>)

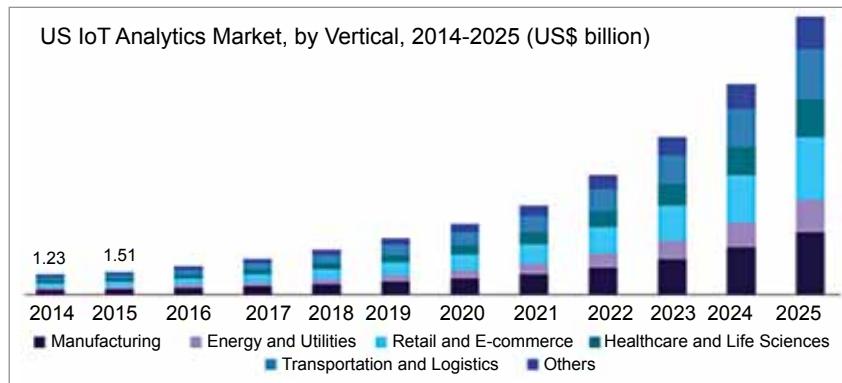


Figure 3: The global (IoT) analytics market size is valued at US\$ 5.7 billion. It is expected to progress at a CAGR of 29.7 per cent over the forecast period (Credit: www.grandviewresearch.com)

analytics is a must to dive deep into data to produce useful insights. IoT and analytics remain connected and grow together.

IoT analytics

IoT analytics emerged for crunching the IoT data. It refers to data tools used to draw values from Big Data generated by the IoT. A lot of companies implement IoT technology to gather data from sensors placed on solar panels, delivery trucks, machinery, and human bodies. A lot of related data comes from gateways, data storage, and network devices. IoT data is streamed at high volumes and high data rates continuously and considered a special case of Big Data.

These streams of data are combined and transformed to present consumable reports. IoT's real-time processing and insights help companies make better marketing and operational decisions. IoT data poses challenges, including data integration, compatibility, and data analytics, which are well addressed using IoT analytics.

Analysts estimate the IoT analytics market to grow at a CAGR of 29.7 per cent by 2025, at a valuation of 5.7 billion dollars. While smart cities, smartwatches, and other 'smart' initiatives are driving IoT adaption, social media, mobile, Internet, and IoT itself are driving the IoT analytics. **END**

By: Rajesh Kumar Adla

The author is the founder of Null Innovation Inc. with over ten years of experience in product development. He has previously worked as an L6 leader at Amazon, delivering fulfilment technologies for Amazon Pantry.

The article was originally published in the August 2020 issue of Electronics For You.

Emerging Trends in the Design of Cognitive IoT

This is an extract from a speech presented by Dr Manjunath Iyer, principal consultant, Wipro, at [iotshow.in](#) 2019. It describes the emerging trends in the design of cognitive Internet of Things (IoT). Rather than just capturing data, cognitive IoT devices can think, understand and accordingly provide data to the user.



Things were there even before the term Internet of Things (IoT) was coined, but these were not connected to the Internet. Today, with the introduction of the Internet, everyday objects (things) have become smart and capable enough to take decisions on their own.

The next step is transformation of these IoT devices into cognitive IoT devices. So, rather than just capturing the data, these devices should also have the ability to think, understand, and accordingly provide data to the user. IoT devices can be turned into cognitive IoT devices by incorporating cognitive computing technologies.

Living IoT devices

Yes, even an animal like buffalo or cow can be transformed into an IoT object. How? Just connect a location sensor to the ear of the animal. Based on the animal's roaming information, the user can know grazing pattern of the animal on a particular day, how much milk it will provide, and thus calculate the amount of profit it will generate.

A living being as small as a honeybee can also be transformed into an IoT object. A small nanosensor can be placed on the body of the insect and, based on its flight path, one can deduce what kind of flowers that honeybee is interested in. This can lead to planting more of such flowers and production of greater amount of honey, resulting in more profit generation.

IoT is not free from issues, however. You cannot put anything and everything, anywhere and everywhere; there are a lot of connectivity issues that you need to deal with.

Let us first understand what an IoT system consists of. It mainly has an object or thing to be connected, connectivity through wireless standards, and the Internet.

The object (things, organisms, etc) to be connected is the main component of the entire IoT system. Connectivity comes in various forms such as Bluetooth, Wi-Fi, Zigbee, NFC, RFID, and many other wireless standards. The data is put on the Internet for storage, processing, and analytics.

When 5G becomes operational, it will be easier to connect many more

devices and put massive amount of data into a network.

IoT is a culmination of such technologies as augmented reality, signal processing, Big Data, mobile computing, object and voice detection, and many more. It can be used in various sectors like banking, energy, industrial, vending, security and surveillance, medical, consumer, telematics, etc. Due to such huge diversity, the chances of it going wrong increases. Therefore the right combination of components should be selected keeping in mind the end result and its applications.

Example of IoT based solution

Consider a mobile phone that remotely operates the lights at home.

This can happen by first registering the objects to be monitored (home lights in this case) on an app installed on the phone. Next, by object identification. Each and every object (light) can be subscribed to the installed application running on the Internet. Through this, one can easily control all the lights.



Communication protocols of IoT

Protocols	MQTT	CoAP	XMPP	SOAP	UPnP
XML based	No	No	Communication protocol for message-oriented middleware based on XML	Relies on XML information set for its message format	No
TCP/UDP	TCP	UDP	TCP	Both TCP and UDP	UDP
IPv4/IPv6	IPv4/IPv6	IPv4/IPv6	IPv4/IPv6	IPv4	IPv4/IPv6
M2M support	OASIS standard	ETSI standard	—	—	—
USAGE	From pervasive devices to servers	Simple electronic devices, Resource-constrained devices	Video transfer	Implementation of Web services	Permits networked devices to seamlessly discover others

Sensors are at the lowest level in an IoT system. These can be temperature sensors, position sensors, velocity sensors, etc. They are connected to the nearest point to track data.

Once the data is collected from the objects through the sensors, it is processed and used in various applications. Application areas include surveillance, clinical infrastructure monitoring, health monitoring, smart transportation, etc.

IoT protocol layers

Now let us look at IoT's working in a bit more detail. An IoT system comprises an application layer, information processing, a network, and sensing and identification. All these four components together make an IoT solution.

The hardware for an IoT system has the following.

Input/output devices: Sensors, actuators, LEDs, relays, motors, linear actuators, lasers, solenoids, speakers, LCD, plasma displays, robots, etc.

Network devices: Modem, gateway, router, satellite, tower, etc.

Processing devices: Grid, cloud, embedded processor, quantum computing, etc.

Memory devices: Cloud memory, flash memory and quantum memory.

The software for an IoT system requires a wireless sensor network (WSN) that has following characteristics:

- Smart and autonomous
- Can be configured automatically
- Self-monitoring and self-healing

- Capable of anomaly detection and tracking
- The event notification service (ENS) of an IoT system acts as a common collector and distributor of events. Events come from multiple heterogeneous sources. Devices can subscribe to specific subsets of events.
- In publish-subscribe software architectural style of ENS, whenever there is a change in an event, a notification is published stating the same. Then, accordingly, messages are generated to be sent to the user. For example, for a voltage change or current change, the event will be interpreted as switching on or off. This will be then conveyed to the subscriber (user) to act upon.

In object as a (Web) service ENS, this is an emerging paradigm. Nowadays, services such as food delivery happen over the Internet. Moving forward, thanks to AI, robots will be aligning themselves with Web technology to provide this service in a faster and efficient manner.

UPnP+protocol for IoT

This protocol is extensively used in present-day high-end IoT solutions. Going forward, it will be required in cognitive IoT because the objects adhering to this protocol can interact better.

UPnP+protocol involves following steps:

Discovery: Device advertises itself within a network and enables discovery.

Description: After the discovery, the control point hits the URLs to get the description.

Addressing: Then the address is auto-generated for ease of use.

Presentation: The focus is on human-machine interfaces.

Eventing: This step is closely coupled with control actions.

Control: Here the device is required to perform some action over other devices.

However, some issues are faced due to the presence of a multitude of IoT devices. In a large distributed system this can sometimes create confusion. Besides, sensors that are placed on a moving object are not made robust enough and at times provide low output.

Issues can arise because of:

- Multiple services, multiple users
 - Heterogeneous services
 - Bursty traffic requests, that is, people sending large amounts of data to the cloud at the same time.
- This causes an overload resulting in system/server crash.

How to handle these issues?

Do not put all applications to use at once. Put some applications to use first so that the moment the device captures data, it also processes it. And divide the data into parts before transferring it to the cloud.



By: Dr Manjunath Iyer

The author has over two decades of experience and has filed about 82 patents, authored 180 research papers and a book.

The article was originally published in the June 2020 issue of Electronics For You.

Internet of Medical Things (IoMT): A Boon for the Healthcare Industry



The advances in medical science are such that now even machines can diagnose symptoms. The Internet of Medical Things (IoMT) can provide a great many benefits to patients and healthcare professionals. This article takes a quick look at its real-time applications, benefits, and the role IoMT can play to fight COVID-19. It also lists popular IoT devices used for the healthcare industry.

The Internet of Things is all around us today, making devices connect with and talk to each other. And now it is also possible to remotely monitor a patient's health with the help of IoT embedded devices, sensors, actuators, etc, thanks to the Internet of Medical Things (IoMT).

IoMT allows healthcare professionals to access personalised information of their patients using a

variety of tools. These tools enable them to track devices, sensors, perform remote patient monitoring, practice telemedicine, provide connected assistance, perform online diagnosis, and eventually help to manage their patients' health more efficiently. Patients can get all the medical services from a healthcare specialist sitting at a remote location.

IoMT also helps to monitor, inform and notify not only patients

but also healthcare providers with actual data to identify issues before they become critical, thus allowing for earlier prevention. However, IoMT isn't intended to replace the current healthcare professionals but to provide them with the data gathered from devices for better diagnoses and treatment plans, as well as to reduce inefficiencies and waste in the existing healthcare system.



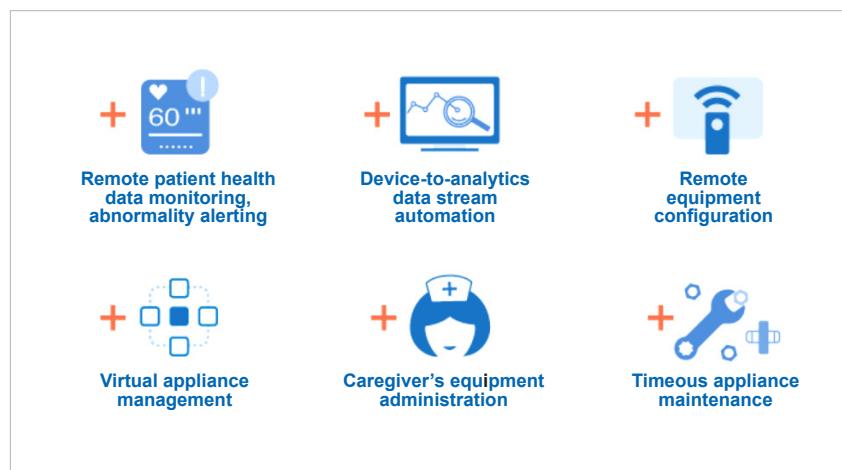


Figure 1: Benefits of IoT in healthcare

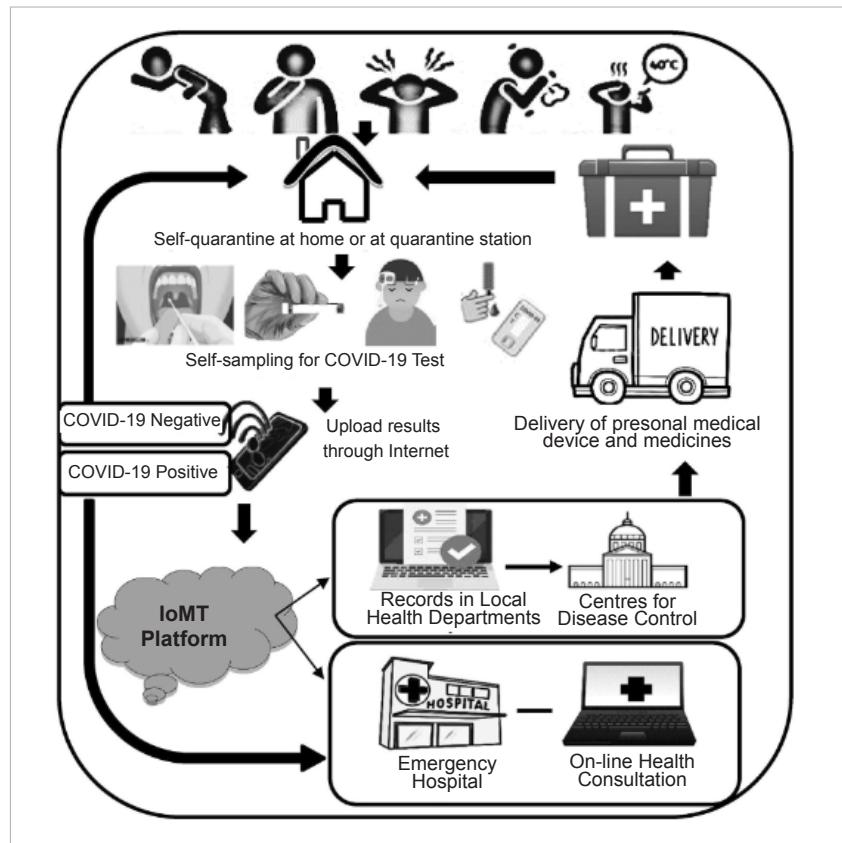


Figure 2: IoMT based solutions for combating COVID-19

The benefits of IoMT are listed below.

- Remote equipment configuration:** IoMT can be used to remotely monitor the patient whenever the physical accessibility of the doctor becomes difficult.

- Improved patient engagement:** IoMT technology also helps the patient in self-monitoring and collecting the necessary information, e.g., a smart thermometer can help track and collect information right from

patients' homes and also assist in telemedicine.

- Identify patient needs and provide tailored services:** IoMT provides the facility to check symptoms in a customised way. As an example, if a patient has diabetes but does not want to measure his heartbeat, IoT gadgets can be customised to do that.
- Improved decision making:** The use of analytics and machine learning in IoMT can improve decision making.
- Helpful for neuro-technology:** IoT devices are capable of reading brainwaves and mood-elevating neuro signals. Thus doctors can use them to track and monitor the mental health of patients from a remote location.
- Improved treatment:** It enables physicians to make evidence-based informed decisions and brings absolute transparency with respect to treatments.
- Infant monitoring:** IoT enabled wearables can monitor and transmit an ill infant's temperature and sleep patterns, which can help doctors to respond accordingly.

How IoMT can help in the fight against COVID-19

IoMT can help healthcare professionals provide medical care for patients who have to be quarantined in the current pandemic. It allows healthcare professionals to access personalised information of their patients using a variety of tools. These tools enable them to track devices, sensors, perform remote patient monitoring, prescribe medicines, provide connected assistance, make online diagnoses, and eventually help manage their patients health more efficiently.

Here are some of the ways in which IoMT can be helpful in combating COVID-19 and

Name	Functionality	Applications
Awair	Air quality sensor that can trigger alerts to the smartphone in case of poor indoor air quality.	Can be used for patients with asthma and allergies.
Fitbit	Wearable health and fitness tracker.	Helps to track fitness goals such as calories burnt, calories consumed, heart rate, feet walked, etc.
GlowCaps	Keeps track of scheduled medications.	Reminds the patient when it is time to take their medicines.
Automated Device For Asthma Monitoring And Management (ADAMM)	Wearable technology that provides a complete solution for tracking, managing, and monitoring asthma symptoms.	Gives the patient real-time data for monitoring asthma.
Leaf healthcare ulcer sensor	I/O medical device with a tri-axial accelerometer sensor.	Senses sitting position, and sends an alert when it's time to get up and move after sitting for long.

Table 1: IoT devices used in healthcare

other related medical health issues around the world.

- **Facilitate diagnosis from a remote location:** IoMT enabled wearable devices can help healthcare specialists treat patients in the self-isolation period from a remote location, and thus reduce their exposure to COVID-19. Based on the need of the patients, IoMT devices can track health problems such as diabetes, high blood pressure, heart failure risks, etc.
- **Thermal imaging:** Thermal imaging technology with sensors mounted on drones can be extensively used at public places such as hospitals, restaurants, office buildings, etc, to track the temperature of incoming and outgoing individuals and send them to healthcare specialists for better monitoring.
- **Assist in social distancing:** IoMT can also be used for social distancing measures. For example, if too many people are gathered in public locations, an alarm can be

programmed to go off and alert the expert team about it.

- **Real-time tracking:** IoMT devices can be used for real-time monitoring of the number of patients who have recovered, number of deaths, and the number of active COVID-19 cases.
- **Reducing the workload of healthcare professionals:** The IoMT devices can help in collecting and analysing data with the use of AI and machine learning. This will help to

reduce the manual workload of healthcare professionals.

Popular IoT devices for healthcare

IoMT devices are available for a wide range of applications in the healthcare industry and play a vital role in improving clinical outcomes. Table 1 gives a list of the most popular IoT devices used in healthcare, which can play a significant role in helping a patient get proper medical services. 

References

- [1] <https://www.iotnewsportal.com/general/iot-devices>.
- [2] <https://iotworm.com/internet-of-things-healthcare-devices/>.
- [3] S. Swayamsiddha and C. Mohanty, 'Application of Cognitive Internet of Medical Things for COVID-19 Pandemic, Diabetes & Metabolic Syndrome: Clinical Research & Reviews', pp. 911-915, 2020.

By: Dr Aditya Bhardwaj

The author is a B. Tech, M. Tech and PhD in CSE. Currently, he works as assistant professor at PEC (a deemed university), Chandigarh. He has experience in cloud computing and Big Data.

Edge Computing: Enhancing the IoT Experience

Edge computing pushes the frontiers of computing, data and services to the extremes of the network, where these are actually needed. It facilitates the collection and analysis of data at its source.



Ihe Internet of Things has brought technology to every household. Consumers see a lot of value in connecting things, and to make sense of the collected data for enhancing their life style. IoT is also widely used in the industrial, automotive and commercial markets, as well as in emerging markets like 5G.

As per Gartner and other sources, the total number of connected devices worldwide today is around 20 billion. This number is expected to be 75 billion by 2025. With so many connected

sensors and devices, users and businesses must be able to analyse the data they produce at the right location and at the right time. However, most of the data management and analysis happens in the public cloud today, which has severe limitations with respect to latency, governance and security. The IoT industry has over the past few years adopted edge computing (and fog computing) to overcome these limitations, by placing the storage and computing nodes at the Internet's edge where the data source is.

Typical IoT architecture and limitations

Typical IoT architecture, as shown the Figure 1, processes the data from the sensors and devices in the public cloud. The IoT gateway, which is a dedicated physical device or a software program, is the connection point between the cloud and the intelligent devices. Its basic function is to pre-process data before sending it to the cloud. It can also autonomously control field devices based on data inputs by sensors. It connects the IoT devices to the Internet using IoT protocols.



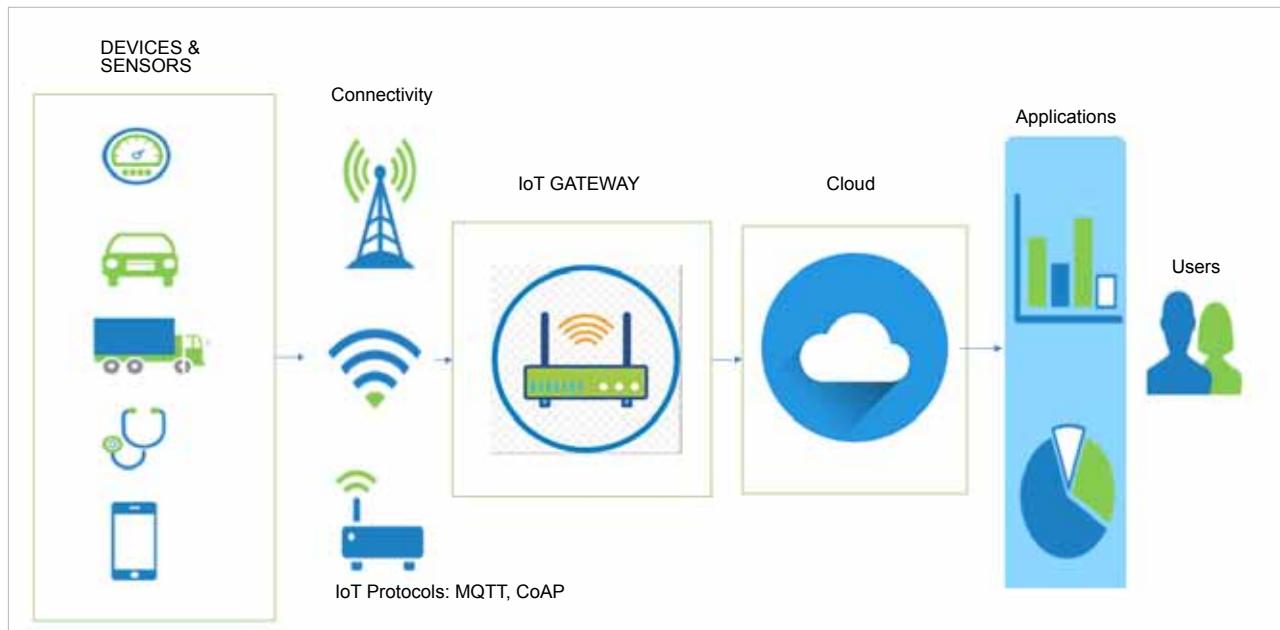


Figure 1: IoT architecture

When the public cloud is used as the data management and analysis platform for an IoT application, the major concerns are the latency involved and the cloud connectivity intermittence, especially for mission-critical and real-time applications like connected vehicles or other autonomous systems. Keeping sensitive data in the cloud is also not something many enterprises want to do. Further, when large volumes of data need to be stored in the cloud, data migration becomes costly.

Edge computing architecture

Edge computing solves the above limitations by bringing storage, analytics and intelligence close to the data source. It allows IT to retain sensitive information on-premise, while still taking advantage of the elasticity offered by the public cloud. The cluster of edge computing devices, which handles network switching, routing, load balancing, security and the audit trail, becomes the local ingestion point for data originating at various sources.

Figure 2 shows the 3-tier edge computing architecture. The first tier is a data layer that includes everything

that is a data source. The second tier is the intelligence layer that spans across the public cloud and the edge layers, wherein the cloud layer does the heavy lifting and the edge layer deals with the production and real-time data sets. The third tier is the insights layer that presents accurate insights delivered by the intelligence layer to the users through rich dashboards.

Fog computing adds a layer between the edge and the cloud. It moves the computing activities to the processors that are connected to the LAN hardware (where the IoT gateway is), which is more distant from the edge devices like the sensors and actuators. Fog computing has the advantage of being able to connect to more devices and process more data; however, it requires more investment than edge computing.

IoT protocols and edge

A few transport-layer protocols that are widely used to form IoT communication links between the devices and the IoT gateway are Bluetooth, Wi-Fi, NFC (near field communication), cellular, Ethernet,

Zigbee, LoRaWAN, WiMax, etc, to name a few. They differ in their range, ease of setting up, cost and bandwidth.

IoT data protocols define the syntax, semantics, synchronisation of data, and error recovery. A few widely used IoT data protocols are Message Queue Telemetry Transport (MQTT), Advanced Message Queue Protocol (AMQP), Constrained Application Protocol (CoAP), RESTful HTTP, Mosquitto (an Open Source MQTT broker), Extensible Messaging and Presence Protocol (XMPP), Data Distribution Service (DDS) for real-time systems, Lightweight Local Automation Protocol (LLAP), Lightweight M2M (LWM2M) and Simple Sensor Interface (SSI), to name a few.

Components of edge and the open source technology stack

The edge gateway (or the IoT gateway) processes the data collected from devices and sensors, and converts it into expected formats. It establishes a bi-directional connection between the IoT devices and the storage and analytics system. It protects information

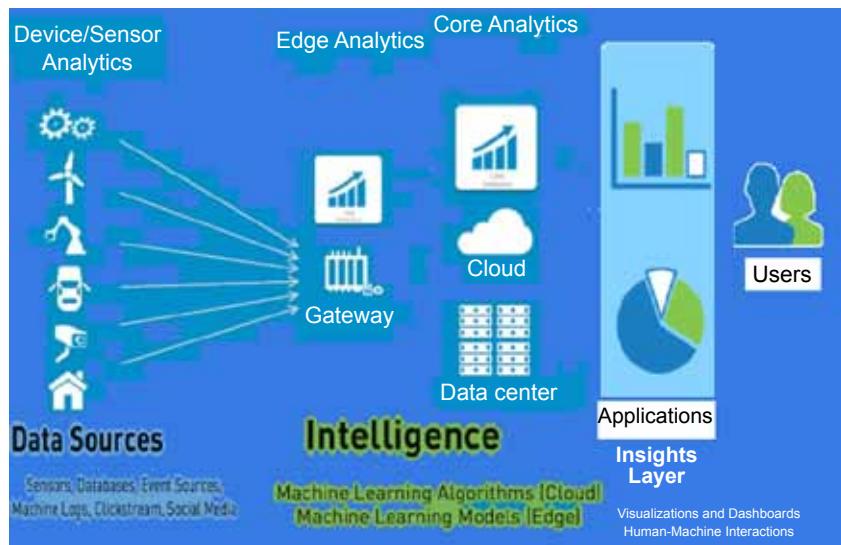


Figure 2: Edge computing 3-tier architecture

flowing in both directions, and prevents unauthorised control of the devices from the outside world. It also facilitates the devices' life cycle management, the phases of which are shown in Figure 3.

The IoT solution consists of Web application interfaces such as Cockpit that provide an interface to administer and configure the devices for management and control. Cockpit can be implemented using front-end technologies like Angular and Node.js.

IoT back-end systems have various components for data ingestion (that use supportive frameworks based on the technology stack, such as Apache Camel and Spark Streaming), for orchestration and processing (that define the business logic and workflows like NiFi and Apache Spark), for modelling and analytics (examples include R, Python, Spark ML, Kibana, Elastic Search and TensorFlow), and for data storage (examples include MongoDB, MySQL, PostgreSQL, Redis, CouchDB and Cassandra).

Major cloud service providers like Azure and AWS have a rich set of technologies and services for aiding IoT, both in the cloud and

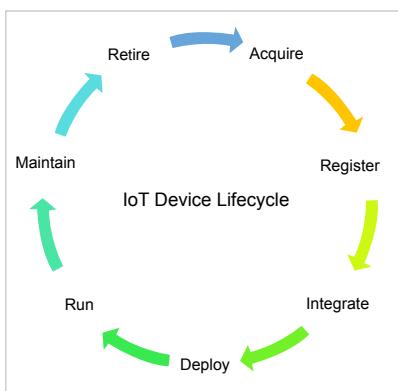


Figure 3: Life cycle of an IoT device

in the edge. Azure IoT Hub, Azure API Management, Azure IoT Edge, AWS IoT and AWS API Gateway are a few examples.

A few open source technology tools, frameworks and projects for IoT systems are LFEEdge, EdgeX Foundry, KubeEdge, HomeEdge, Kura, Kapua, Spring Boot, Eclipse Hono, Kaa, OpenIoT, DeviceHive, OpenThread, etc, to name a few. Roof Computing is a standard from IEEE to drive edge computing standardisation.

Open source security and IoT

More connections mean more pathways and back doors that can be exploited by hackers. Open source components are vulnerable to attacks as the developers of these components are not security professionals. In order to use open source for IoT on the edge, the software needs to be analysed end-to-end because the code on the device is no longer standalone. Once scanned, all vulnerabilities should be addressed to create robust systems.

Edge computing has enriched the IoT world by giving the users of smart devices a complete sense of control and security, thus improving the confidentiality, integrity and availability of IoT data. Edge can be compared to the hybrid cloud, which is very popular and is being increasingly adopted by enterprises moving their businesses to cloud. Edge services and infrastructure, too, will see increased demand in the coming years, not only in the IoT domain but also in other emerging markets. **END**

References

- [1] www.iot-now.com
- [2] www.openstack.org
- [3] Forbes Article: Edge Computing – Redefining the Enterprise Infrastructure
- [4] Edge computing articles in OSFY

By: Anita Nair

The author is a software and technology enthusiast who works at EY, India. Her areas of interest include cloud computing, cyber security, software design patterns, DevOps, computer networking and infrastructure, emerging technologies and research. She is a senior member of IEEE and a life member of CSI. She maintains a technical blog at <https://mindovertechnologyblog.wordpress.com/>.

Have Some Fun Doing These Useful Home IoT Projects with Raspberry Pi



IoT projects can be fun and a great learning experience for budding programmers.

Due to the pandemic, most of us are still stuck at home. This can be a great opportunity to dive into some DIY projects to enhance our programming and tech skills. In this article, we look at a few useful IoT projects that are fun and can be done from the comfort of your home using a Raspberry Pi along with tools and gadgets that pretty much everyone owns.

Internet of Things (IoT) is defined as devices connected over the Internet that are able to communicate or send data between each other. IoT is a very broad area and its capabilities are limitless. A range of skills is needed to employ and actualise IoT technologies, and most of these skills are hard to come by as the technology is still maturing and has a lot of vulnerabilities. Having said that, learning how to do simple DIY projects using IoT is a great place to start for any technology enthusiast, specially programmers.

Raspberry Pi

The Raspberry Pi is a small, low-cost computer that can be used for, among other functions, DIY (Do It Yourself) IoT projects. It is a plug and play computer with an open source operating system named Raspbian, which is a distribution of Linux. It was initially designed to help kids at school learn the basics of programming at a low cost. However, since then, Raspberry Pi has become a favourite of many open source and IoT enthusiasts around the world, who have pushed the functionality and usability of this

device to great extremes. Added to that, Raspberry Pi also has a huge and active community online to help answer your questions and offer support. So, without further ado, let us jump in and discuss three different fun-filled and useful DIY projects that you can do today in the comfort of your home, just by using a Raspberry Pi and other simple tools and gadgets commonly owned by most.

Domoticz

Domoticz is a home automation system that lets you monitor and configure various devices like lights, switches,



and other sensors and meters. By using it, you can easily manage your home devices with your mobile phone. Domoticz can be installed and used on Raspberry Pi. Let us take a look at how this is done.

What you need

- Power supply and source
- MicroSD card
- Raspberry Pi
- Ethernet cable
- Microphone

 **Note:** Set up the Raspberry on a static IP address.

- i. Upgrade all the packages on the Raspberry Pi by running the following commands:

```
sudo apt update
sudo apt upgrade
```

- ii. Install Domoticz in the Raspberry Pi by running the install script from its website. Run the following command:

```
curl -L https://install.domoticz.com
| bash
```

After the script run completes, press the *Enter* key.

You will then choose the services you want Domoticz to operate on. For this tutorial, we will leave both HTTP and HTTPS on. But some people prefer to disable the HTTP so that they can use the more secure HTTPS. Press *Enter* to continue.

- Choose the port you want HTTP to run on, although this is usually by default 8080; so just press *Enter*. And do the same for the HTTPS port, which is 443 by default.
- iii. Choose the Domoticz installation folder, which usually by default is the *home* folder. Press *Enter*.
 - iv. The installation process is complete at this point and you will be shown two IP addresses, which you

should take note of. Press *Enter* to complete the installation process.

Configuration of the Domoticz Web interface

Get the static IP address we assigned to the Raspberry Pi above. Through a Web browser, run:

[http://\[IPADDRESS\]:8080](http://[IPADDRESS]:8080)

At *IPADDRESS* use the static Raspberry IP address.

At this point you will have access to the Domoticz Web interface and can now monitor your various connected home appliances.

Raspberry Pi Bluetooth speakers

This project is a little bit more difficult if you do not have the right software tools; it is for this reason that we will use the Balena image (*balenaSound*) which will simplify our work. This can enable audio from Bluetooth and Spotify, as well as other synchronised audio.

What you need

- Power supply and source
- MicroSD card
- Raspberry Pi
- Ethernet cable
- Bluetooth adapter if it is not already available

Setup steps

Download *balenaSound* image for use with Bluetooth. You have to have a Balena account to proceed.

- i. Create an application with the Balena Web interface.
- ii. On the screen that appears, choose the *Default Device Type* and select the Raspberry Pi that you will use as a default device. After that, click *Create and deploy*.
- iii. On the page you are taken to with the devices you have deployed, click on *Add device* to add Bluetooth speaker software, as it is not on the list.

- iv. A pop-up that will allow you to create a balenaOS image for the device you added in (iii) above appears. Select the device type; then choose the version of the balenaOS image to use. For this purpose, we choose the *Production* option. Another option is the *Development* option, but this is used for troubleshooting purposes.

To use Wi-Fi, click the toggle under *Network Connection*; but you must also specify its SSID and passphrase.

- v. Now download balenaOS.

Adding the balenaOS to your Raspberry Pi

To add the balenaOS to your Raspberry Pi, we will use the Etcher tool to write the *balenaSound* image to the Raspberry Pi's SD card.

- i. In the Etcher software, click on the *Select Image* option. From the file dialogue that appears, select *balenaSound* image to write to the Raspberry SD card.
- ii. Choose the image you want to write the image to.
- iii. Click on *Flash* to write it to your SD card. Note that the drive you choose will be wiped.

After completion, you can now plug the drive into the Raspberry Pi to run.

Checking the status of your Raspberry Bluetooth speakers on Balena

- i. Go to the Balena dashboard page and click the *balenaSound* application.
- ii. A list showing the devices running on this application will appear, and from it you can check the status of yours.
- iii. There is a *services* section from which you can see the Bluetooth audio service, Spotify and Airplay. From the interface, you can control the services by taking actions such as *start*, *restart* or *stop*.

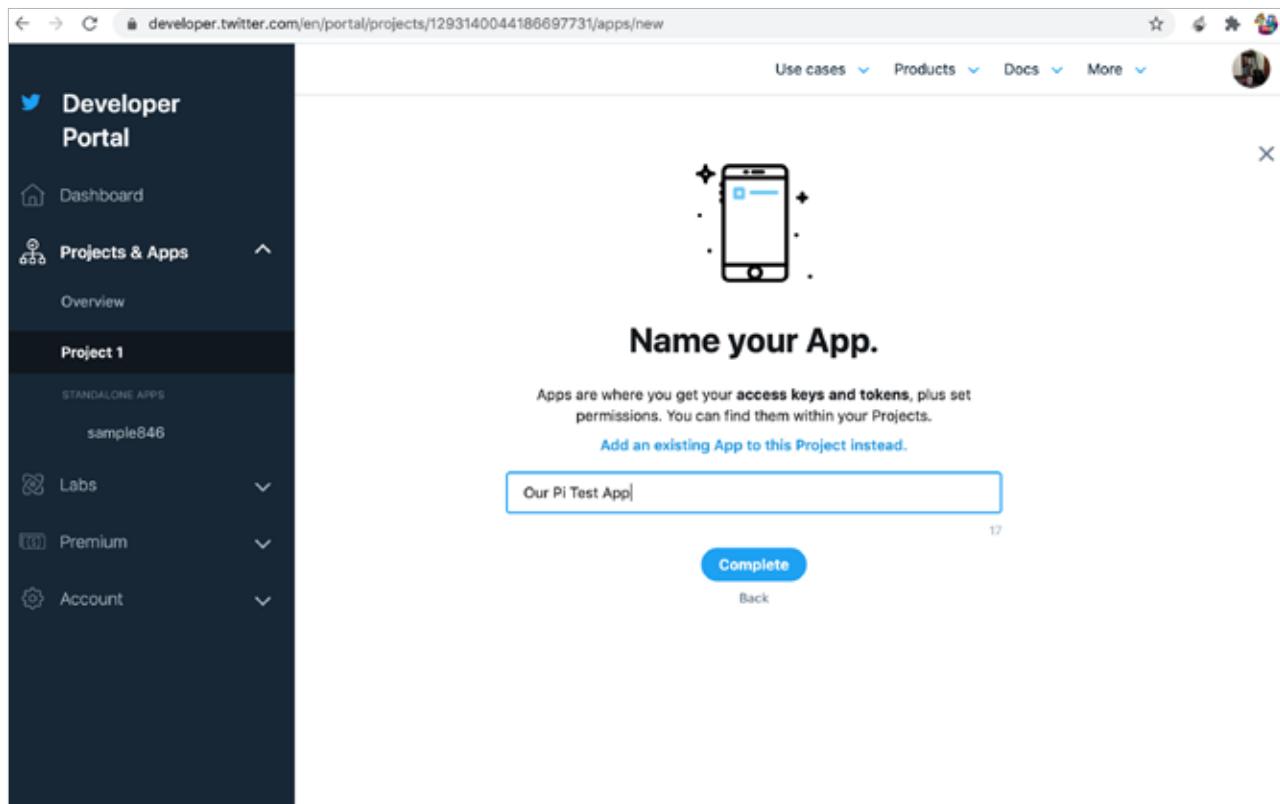


Figure 1: Fill in the credentials on the *Twitter New App Page*

Connecting your Raspberry Bluetooth speakers

There are three methods to play music using the speakers: Bluetooth, Airplay and Spotify connect.

Through Bluetooth: Scan for the available Raspberry Pi Bluetooth speakers. Choose the one that appears as *balenaSound* and has a unique ID. With this, you can play music from your phone through the Raspberry Pi.

Through Airplay: BalenaSound has a built-in support capability for Apple AirPlay. This is used to send music from one device to another over a network. Find *balenaSound AirPlay* from the AirPlay devices list, given that it is followed by a unique ID.

Through Spotify Connect:

Follow the same procedure as that for Bluetooth above by looking for *balenaSound* that has a unique identifier and choosing it.

Raspberry Pi Twitter Bot

This third project is a simple Twitter Bot built using Python as the programming language. The example below will show how to write the script while also implementing Twython.

What you need

- MicroSD card
- Raspberry Pi 2/3
- Power supply
- Ethernet cord

Setup steps

1. **Register a Twitter App:** In order to use the Twitter API, you first have to register the account as a Twitter application. This will allow you to use the REST interface, which then allows you to interact with Twitter. After registration, go to the *Twitter New App* page and set the name, description and website to what you want, and then go to *Create Your Twitter Application*.

2. On the page you are taken to next, click on the *Keys and Access Tokens* tab. While here, copy both the consumer key (API Key) and the consumer secret (API Secret); these will be used to connect to the Twitter API. Ignore the other fields as they are not necessary; then click on *Create my access token* button.

3. The page will refresh, after which you should take both the *Access token* and *Access Token Secret* values at the bottom of the page.

4. **Setting and writing the Twitter Bot for Raspberry Pi:**

- i. Make sure Raspberry is up-to-date by running these commands:

```
sudo apt-get update
sudo apt-get upgrade
```

- ii. Install the needed tools to enable the Raspberry Pi Twitter Bot to run. As discussed earlier, we will

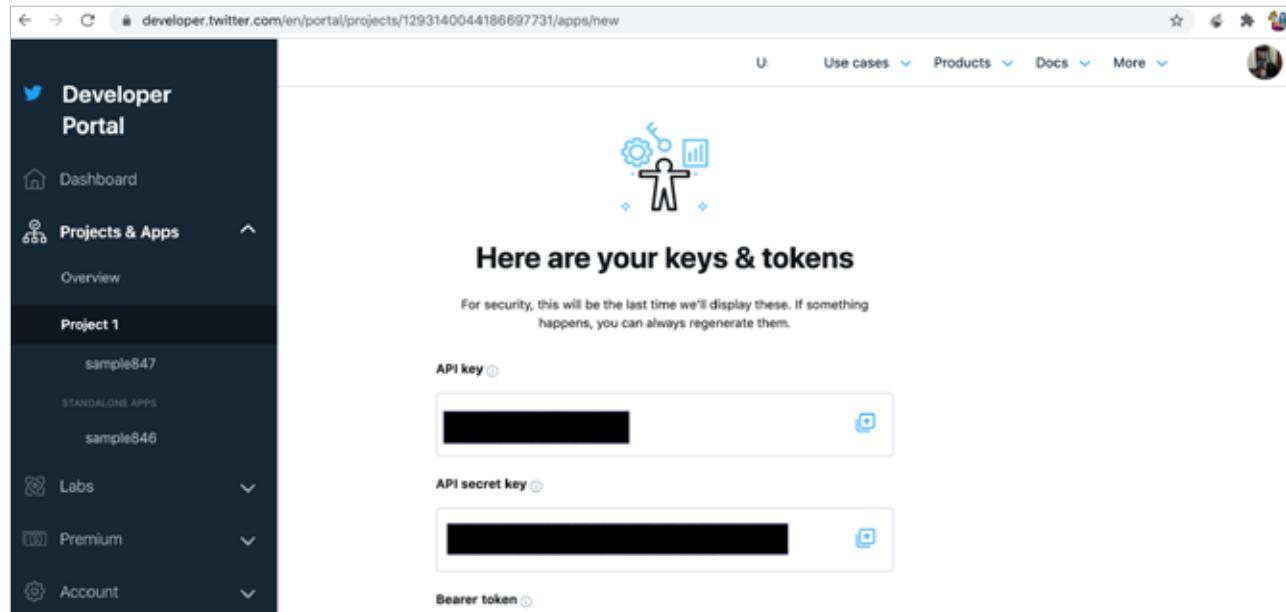


Figure 2: Adding Access Token and Access Token Secret

be using the *Twython* package to interact with the Twitter API. Run the following commands:

```
sudo apt-get install python-setuptools
sudo easy_install pip
sudo pip install twython
```

iii. After the installation of the *Twython* package above, create a directory to store your script. Run the following commands:

```
mkdir ~/twitterbot
cd ~/twitterbot
sudo nano TwitterBot.py
```

iv. Now write the final piece of code and run it:

```
import sys
from twython import Twython

#Define our constant variables, this is
#all the data we wrote down in the first
#part of the tutorial.
CONSUMER_KEY = '*****YOUR
DATA*****'
CONSUMER_SECRET = '*****YOUR
DATA*****'
ACCESS_KEY = '*****YOUR
```

```
DATA*****
ACCESS_SECRET = '*****YOUR
DATA*****'
```

```
#Create a copy of the Twython object
with all our keys and secrets to allow
easy commands.
```

```
api = Twython(CONSUMER_KEY, CONSUMER_
SECRET, ACCESS_KEY, ACCESS_SECRET)
```

```
#Using our newly created object,
utilize the update_status to send in
the text passed in through CMD
api.update_status(status=sys.argv[1])
```

Replace the values of *CONSUMER_KEY*, *CONSUMER_SECRET*, *ACCESS_KEY* and *ACCESS_SECRET* above with the values you had got earlier. This code above is to allow communication and writing of text to Twitter. Save and file the code by pressing *Ctrl + X* followed by *Y*, and then pressing *Enter*.

v. Make the above script executable to enable passing of text to the Python file as an argument. Run:

```
sudo chmod +x TwitterBot.py
```

vi. You can now test the Twitter Bot by running the following command and checking your Twitter account to see if the tweet was sent:

```
sudo python TwitterBot.py "This is a
test tweet from PiMyLifeUps Twitter
bot"
```

In this article, we have taken a look at three useful DIY projects that are also fun and can be done at home with ease. I have shared these projects with the intention of only getting your feet wet with the innumerable applications of the Raspberry Pi. I hope these projects become a good entry point for you to advance to more interesting and fun-filled projects in the field of IoT. 

 By: Mir H.S. Quadri

The author is a scholar and researcher in the fields of artificial intelligence and machine learning. He has worked on multiple projects using a wide array of frameworks. He is also a FOSS enthusiast, and actively contributes to several open source projects. You can find his works on codelatte.site.

Horizontal IoT Platforms: An Overview

The Internet of Things (IoT) along with Big Data is captivating the digital world. A horizontal IoT platform encourages a large number of clients to work with a typical framework, thus enabling rapid growth and innovation in businesses. By making the gateway and cloud resources have open functionality, innovators can concentrate their efforts on creating devices and services.



The backbone of IoT is connectivity; hence, an IoT platform can be as simple as software infrastructure that handles connectivity between physical objects. Due to a rise in the number of connected endpoints and smart devices, the IoT market is placing a huge demand on platforms that provide real-time intelligence, machine-to-machine interaction, cross-industry applications, multi-vendor interoperability and end-to-end security. Hence, the platform may be furnished with entities such as artificial intelligence (AI), machine learning and augmented reality. IoT basically consists of the following building blocks.

- **Connectivity:** All IoT platforms

have a connectivity layer that provides a duct for data acquisition, remote access and control.

- **Device management:** Provisioning tasks, device registry, device identification, device modelling, remote administration, software updating, authentication of cloud/backends, centralised log collection and management, fault tolerance, troubleshooting, diagnostics and remote reboot, are all tasks that are handled by the device management module.
- **Data management:** The data management block refers to the expansion, execution and administration of architectures,

software and disciplines that control, defend, deliver and improve the value of data and information assets.

- **Application enablement:** Application enablement provides action management to expound rules, and activates the automation of tasks and services.
- **Cloud back-end:** Cloud back-end has the ability to handle extensive amounts of data and is cost-effective in comparison with on-premise infrastructures. Cloud computing is deemed an essential part of commercial IoT.
- **Security:** We need to provide lots of security, trust, privacy and identity management across endpoint devices, the network and the cloud in IoT platforms. This is performed by the Security Block.

Types of IoT platforms

IoT platforms can be categorised into different genres based on the unique types of services provided to end users. Platforms for smart wearables and home automation aim at the consumer market, and on unidirectional, data-to-cloud connectivity. On the other hand, Industrial IoT (IIoT) customers from sectors such as military, manufacturing, energy, automotive, agriculture, aerospace and banking are concerned with embedded software, tailored applications, device management, data management, interoperability, security, reliability



responsiveness and high frequency bi-directional connectivity, etc.

Commercial IoT platforms are classified into five sections.

(i) End-to-end IoT platforms:

These are full-stack platforms performing all services, including connectivity, protocol interoperability, data-structure interoperability, device management, device firmware/software upgrade management, rules engine, actions engine, ticketing management, data visualisation, analytics, multiple applications and accounts, applications templates, applications reuse and duplication, user management, dashboard building and customisation, reports, external interfaces, security, and data storage.

(ii) Connectivity management platforms:

These are elementary IoT platforms that offer solutions for connectivity orchestration, connection management and provisioning of communication services for connected IoT devices.

(iii) Cloud storage platforms: Cloud based IoT platforms are the measurable enterprise grade back-end for data management of IoT applications.

(iv) Device management platforms:

These offer device provisioning, remote administration, software updation, firmware upgrades, security patching, remote control of devices, monitoring and reporting about specific metrics associated with IoT assets.

(v) Analytics platforms: Analytics infrastructure runs analytical applications, Big Data processing, contextual mining and machine learning algorithms. Analytics platforms provide statistical and optimisation tools to extract valuable information from structured and unstructured data.

Here is a list of a few popular IoT platforms:

- Amazon AWS IoT
- Google Cloud IoT
- Microsoft Azure IoT
- Oracle IoT Cloud
- IBM Watson IoT

Open standards and a horizontal architecture

Developing open guidelines enables a horizontal architecture for IoT arrangement suppliers and gadget providers to expand on. A standard, IoT industrywide design aids in similarity and start-to-finish coordination since arrangement suppliers, IT associations and gadget makers can anticipate regular prerequisites when creating and actualising arrangements.

Horizontal architecture provides open connectivity, improved security, scalability and flexibility. It provides end-to-end integration, more solutions (i.e., normalisation, analytics and streaming), better privacy and security, higher quality and lower cost. IT organisations, in particular, are at the heart of enterprise IoT implementations and can help lead the transition to standards-based horizontal architecture.

Different horizontal platforms

Canopy

Canopy simplifies the IoT cloud by acting as a cloud relay between IoT devices and applications. It enables rapid development of IoT solutions for consumer, commercial and industrial markets. Canopy is a collection of software tools and libraries that simplify the development of cloud-enabled products. By ‘cloud-enabled product’, we mean any physical device with a connection to the Internet that can be monitored and/or controlled over the Web. Such products may include consumer appliances, industrial equipment and other IoT devices. A cloud-enabled product needs software

to run in several places -- on the device, in the cloud and within client-side applications. Canopy contains sub-projects for each of these:

- Canopy Embedded Development Kit (EDK) software runs on the device.
- Canopy Cloud Service (CCS) software runs on the server-side.
- Canopy Application Development Kit (ADK) software runs within an application — for example, on a smartphone or in a Web browser.

Chimera IoT

Chimera IoT provides real value to the world through IoT using anonymised data, crowd sourced remediation, and sophisticated rules based analytics integrated with AI and validated through a human feedback loop. Chimera IoT uses six technology products and services including HTML5, Google Analytics and Google Fonts. The six technologies it uses for its website include Amazon, Google and Sectigo SSL.

DeviceHive

DeviceHive is free open source software that helps in smart gadgets communication and its management. It comprises a communications layer and control software with multi-platform libraries, which allow users to improve smart energy, robotisation, remote sensing, remote control, telemetry, and more. DeviceHive also provides REST, Web socket APIs by default plus MQTT API as an extra plugin. All communication is performed by means of JSON messages.

Nimbits

Nimbits is a hybrid cloud server. This horizontal platform is built on constrained embedded systems by suggesting IoT related services. It can run on Amazon EC2, Google App Engine, Ubuntu Linux KVM based virtual machine and also

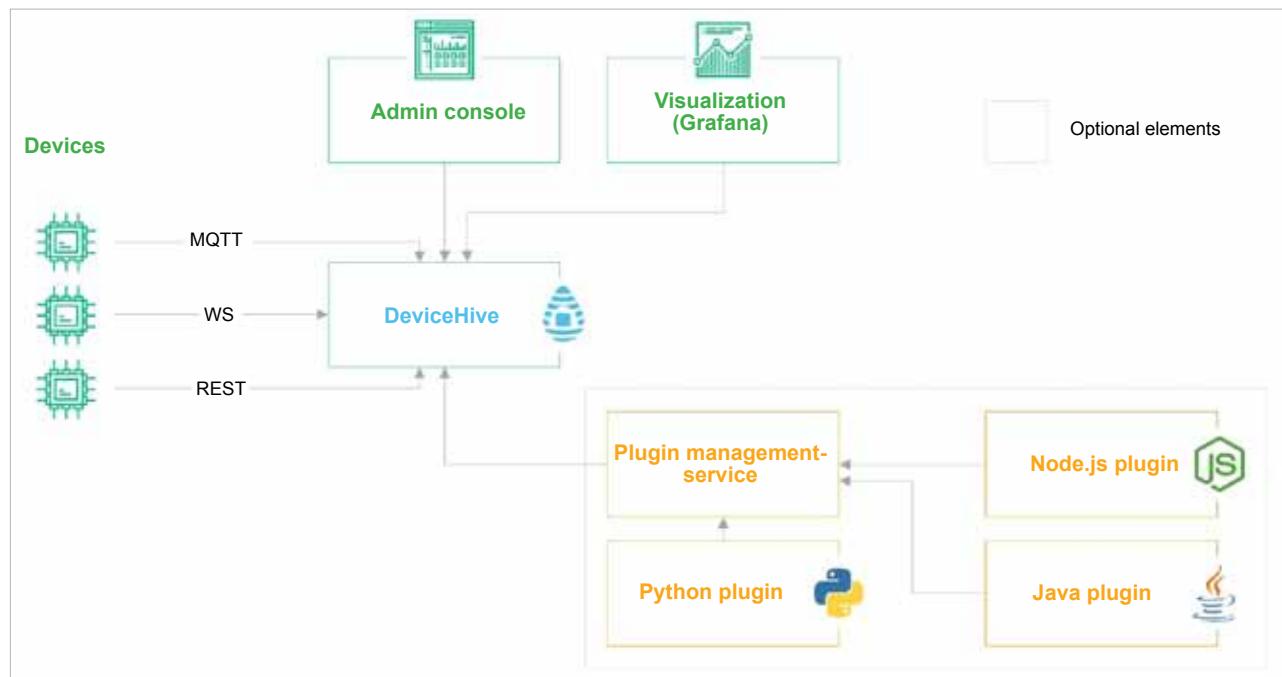


Figure 1: DeviceHive

Jetty (i.e., J2EE Web server) based frameworks. Nimbots adopts open data table services such as Google Data Table Format to use in graphing, and includes them in spreadsheets and also third party analytics tools.

Open Source Internet of Things (OSIoT)

Open Source Internet of Things (OS-IoT) is a free open source software library that facilitates the advancement of IoT gadgets by having an interface with an open, interoperable ecosystem.

The OSIoT library gives device-side (for example Application Entity in oneM2M terminology) support for fundamental oneM2M defined functions. The library offers help for the oneM2M network, and protocol functions permit app developers to collaborate with the system over a resource-oriented API.

prpl Foundation

The prpl Foundation is an open source platform. It is a network driven, community-oriented and non-benefit establishment that endeavours to enable

the security and interoperability of embedded devices.

RabbitMQ

RabbitMQ is the most widely used open source message broker. It is lightweight and easy to send on-premise and in the cloud. It can meet high-scale, high-accessibility necessities by deploying in distributed and federated configurations.

SiteWhere

SiteWhere is an industrial strength open source application enabled platform for the Internet of Things. It delivers a multi-tenant microservice based infrastructure, which has the key features required for IoT applications to be built and deployed.

Webinos

Webinos is an application platform that is Web based. It enables a developer to

approach native-like competence by means of JavaScript APIs. In addition, it features a set of interoperable protocols that can permit a device to access and use the services of another device remotely.

Yaler

Yaler is a relay service that is simple, open and scalable. It entitles secure exterior access to Internet-connected embedded frameworks and IoT gadgets that are beyond a router or a firewall system.

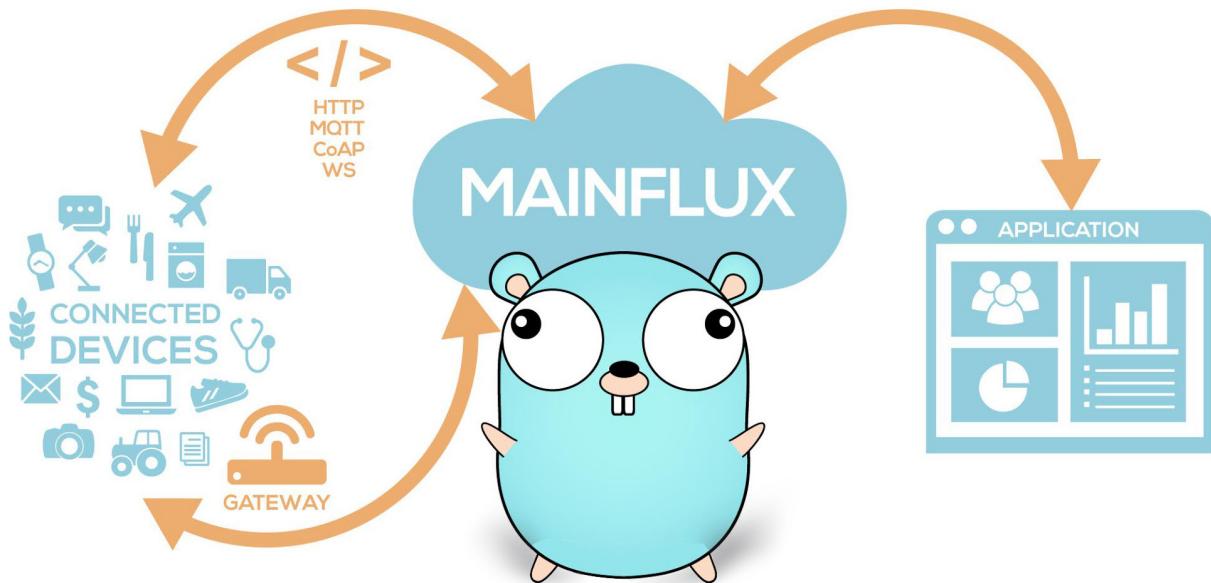
Connectivity issues can be triggered by firewalls and Network Address Translators (NATs) when installing embedded systems and products based on IoT in the real world. The horizontal approach can enable rapid growth and innovation in businesses by allowing multiple service providers to work on a single platform. However, it still has to acquire significant traction. 

 By: Dr S. Balakrishnan

The author is the head of the Department of Computer Science and Business Systems at Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu.

Exploring Mainflux and Its Usage

Mainflux is a highly secure, open source and patent-free IoT cloud platform developed in the GO programming language. It is based on a set of microservices. It accepts connections over multiple network protocols like HTTP, MQTT, Websocket, CoAP, etc, making a seamless bridge between them. Mainflux is used as IoT middleware, on top of which advanced IoT solutions can be constructed.



Mainflux is used as the IoT middleware for constructing complex IoT solutions. It makes use of south bound API to connect devices and northbound API to connect applications, and then it permits message routing between them. The whole bottom-to-top system (device, Mainflux platform, and application) bundled together is called a ‘vertical solution’.

From Figure 1, we can understand that Mainflux works as middleware that does a lot of heavy lifting. The product designers have only to focus

on connected devices (physical design, firmware, etc) and the end user application (GUI that controls the device), as the Mainflux block can be reused. Mainflux sends messages between connected devices and applications in a highly secure manner, also offering many other services needed for these devices and applications. It offers multi-protocol device-agnostic message relay with distributed time-series data storage, and multi-user, multi-tenant device and application management middleware.

Architecture

Mainflux works as middleware, which means it employs a set of servers that provide several functionalities and services used in developing IoT applications.

The features of Mainflux are:

- Messaging bridge (HTTP Sender, MQTT and WS, CoAP)—relays messages between devices and applications
- Systems manager (Manager)
- Device manager (Device)—accepts device connections on southbound interface



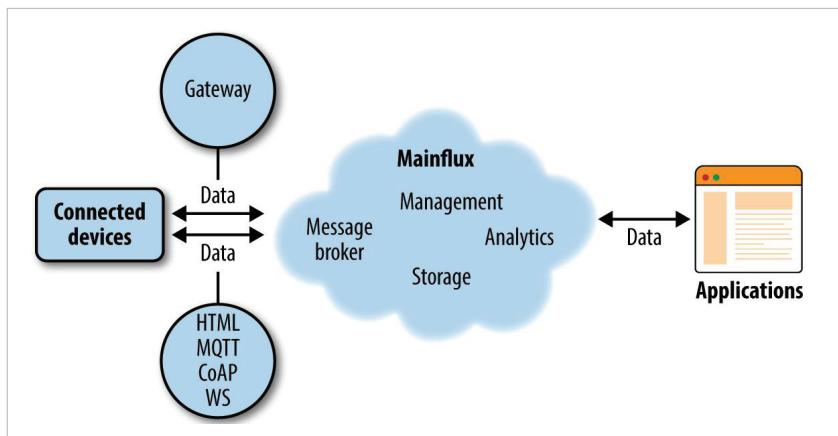


Figure 1: Location of Mainflux inside the IoT system

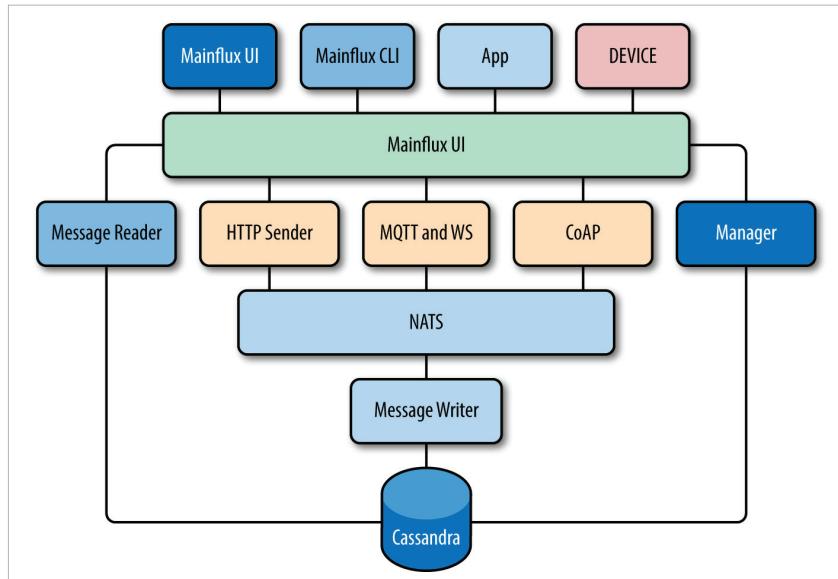


Figure 2: The key constituent elements of Mainflux

- Application manager (App)—accepts application connections on northbound interface
- User manager (Mainflux UI, Mainflux CLI)—provides user management for the applications
- Time-series storage engine (Message Reader, Message Writer, Cassandra)—stores and queries measurement data points in the time-series format
- Complex Event Processing (uses the system event bus NATS)—consumes the incoming time-series streams, and can automate triggers and actions based on a configurable set of rules.

Apart from messaging, Mainflux contains three main domains: user, thing and channel.

User represents the real user of the system. A user can access the platform through the access token, which can be obtained via an email id and password. Once logged, users can access or manage the resources (things and channels) by CRUD operations, and can define access control policies by connecting them.

Thing represents the applications or devices connected to Mainflux, which use the platform for message exchange with another device.

Channel represents the way through which a message is sent; simply put, we can say it is a communication channel.

Installation

Before installing Mainflux, install the prerequisites given below:

- Docker
- Docker Compose

Once the above components are installed successfully, the next step is to download the source code. You can do this in two ways — with or without the Git tool. I prefer to do it with the Git tool as it will download all the source code in the appropriate folder. But if you do this without the Git tool, you will have to download the Zip file from the GitHub repository, and extract the source code to the destination directory on your machine. Once it is done, you are ready for the next step.

We will now use the terminal to navigate to the folder where we have downloaded or extracted the Mainflux source code. For example, for my machine I used the `/home/user/` directory. If the Git tool is used in the previous step, then there should be a `/home/user/mainflux` directory as your path. Else, if you have used the Zip file and extracted it to your machine, then a folder might be created with the name as `mainflux-master`. Navigate to that source code directory. The next step is to start the Mainflux core services by executing the command:

```
docker-compose -f docker/docker-compose.yml up -d
```

Assuming that the above command is executed successfully, the Mainflux IoT platform should be up and running.

Some points to note

- Docker composition is defined under your project `root/docker/docker-compose.yml` file. You can customise this file according to your specific needs.

Mainflux				Things	Channels	Logout
Id		Name	Type	Payload		
1		MyAPP1	app			
2		MyDev1	device			

Figure 3: Dashflux home page

- The most likely problems to occur are port conflicts. If any of your applications is using a port that is needed to run Mainflux services, then you need to free that port or modify the default port from *docker-compose.yml*.
- Another common problem faced is failure to allow all the ports in the firewall configuration.

Using Mainflux

Mainflux is basically about messages, for which it provides a simple UI client known as Dashflux. As said earlier, to login for the first time you need to register by providing a valid user name and password. Once logged into the system, you can manage resources with the help of CRUD (create, read, update and delete) operations, and can define the access control policies. After login, the user is redirected to the home page (Figure 3).

You can see three buttons on the top right corner in the Dashflux home page. The *Things* button will navigate to the things dashboard, *Channels* to the channels dashboard and *Logout* will log you out. Clicking on the plus symbol will open the dialogue for adding a new thing or channel. Once a channel is created, the user can edit

it and connect or disconnect devices. Remember one thing can be connected to more than one channel, and vice versa. Mainflux uses Rest API — all the operations can be done by using various command line tools or any HTTP API development tool.

Figure 3 can be divided into two parts: core platform components and optional ones. Core platform components are NATS, protocol adaptors and *normalizer*. *Writer* and *database* are optional. Here, ‘thing’ represents an application or device that uses Mainflux for message exchange. As the need to use Mainflux is different for every company, the protocol adaptor is used to support various types of protocols like MQTT, WebSocket and HTTP. For every protocol used, there is a corresponding protocol adaptor. The job of the adaptor is to transform a protocol-specific message to a suitable format, i.e., Mainflux message. Then comes NATS, which is an open source scalable messaging system Mainflux uses for message exchange within a platform. Once

published to NATS, the message will be transferred to *normaliser* service. The job of *normaliser* is to convert the message to the SenML (Sensor Markup Language) format. *Normaliser* then forwards the converted message to NATS for further processing, to check whether it is in SenML or not. Only SenML formatted messages will be written in the database.

Security

Mainflux is a highly secured system. It has dedicated authentication and does not allow access to unauthorised devices and applications. All the messages and the network traffic are encrypted by the latest security standards such as TLS v1.3.

Mainflux is licensed under the Apache License 2.0. It is a lightweight and scalable platform. As all the services are deployed as a Docker container, the whole platform setup and installation takes only three steps. We can also customise whatever service or feature we like, and contribute to the Git repo if we wish to.

By: Neetesh Mehrotra

The author works at NIIT Technologies as a senior test engineer. His areas of interest are Java development and automation testing.

Managing Project Source Code Development Efficiently with Git

The first part of this article introduced Git, its workflow, installation and basic commands. This second part elaborates the collaboration aspects of Git, including branching and merging, which help with parallel, non-linear development in projects and avoid disruption in the main line of development. Working with remote repositories is also covered.



 Branching in Git is analogous to branches on a tree. A Git branch is an independent line of development emerging from the main development line, just like branches coming out of the trunk of a tree. This feature at its core enables the collaboration of a team on a single project by letting developers work on feature developments, bug fixes or hot fixes without disturbing the main development line on their own branches, and merging them whenever necessary.

In most version control systems (VCS), branching is an expensive operation with respect to both time and space. Git branches are extremely lightweight, making operations like creating branches and switching back and forth from one to the other very quick. This feature sets Git apart from the other VCS.

Creating and using branches

`git branch <branch-name>` is used to create a new branch. It creates a new pointer from the current branch. When the branch is put to use, it does affect the current branch whenever any changes or commits are made in the repository. `git checkout <branch-name>` is used to make the HEAD point to the `<branch-name>`. In other words, this updates the files in the repository to match the version stored in the branch, and tells Git to record any changes from now on to the branch.

Both these commands can be combined in a single command like `git checkout -b <branch-name>`. Let's create a feature branch and record some changes in it, as explained in Figure 1. We will continue working on the same repository that was created in the first part of this article carried in the August 2020 issue of OSFY.



We can easily switch back to another branch (in this case master branch), work separately on it and commit the changes, which is as shown in Figure 2.

```
(base) ekta@git-demo$ git checkout -b feature
Switched to a new branch 'feature'
(base) ekta@git-demo$ echo "branching in git" > branch_file.txt
(base) ekta@git-demo$ git add branch_file.txt
(base) ekta@git-demo$ git commit -m "First commit on a branch"
[feature d94db016] first commit on a branch
 1 file changed, 1 insertion(+)
 create mode 100644 branch_file.txt
```

Figure 1: Create and check out a new branch

```
(base) ekta@git-demo$ git checkout master
Switched to branch 'master'
(base) ekta@git-demo$ echo "new line in file1" >> file_1.txt
(base) ekta@git-demo$ git add file_1.txt
(base) ekta@git-demo$ git commit -m "master branch commit"
[master 5626b79] master branch commit
 1 file changed, 1 insertion(+)
```

Figure 2: Check out the *master* branch

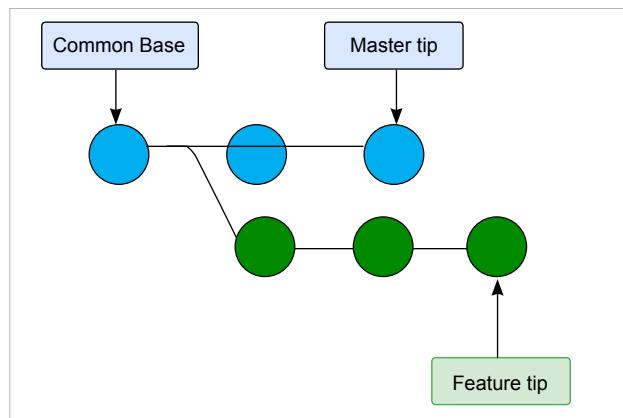


Figure 3: Snapshots of branches before *merge*

```
(base) ekta@git-demo$ git checkout feature
Switched to branch 'feature'
(base) ekta@git-demo$ git merge master
Merge made by the 'recursive' strategy.
 file_1.txt | 1 + 
 1 file changed, 1 insertion(+)
```

Figure 4: Command for merging *master* branch to *feature* branch

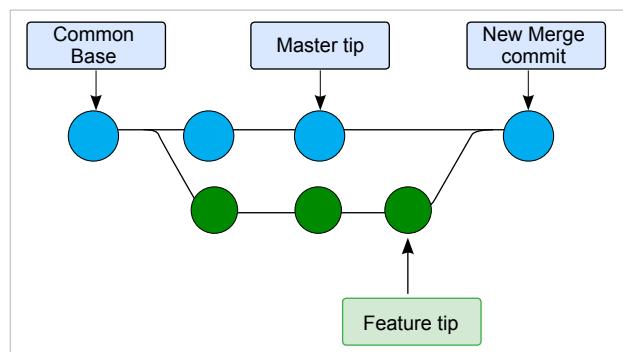


Figure 5: Snapshots of commits after merge. A new *merge commit* is created

Integrating the branches

There are two ways to put the work done in branches onto the main line of development — merge and rebase. Both ways take the series of commits from the target and current branch, and combine them into one unified history, but both do it in different ways.

Based on which branch needs to be updated from which one, the branches can be designated as source branch and target branch. In this scenario (refer Figure 3) we consider *master* as source branch and *feature* as target, assuming that work being done on the *feature* needs to be updated with work on the *master*.

Merge: Merging is done after checking out the target branch and then using *git merge <source-branch-name>*, which is shown in Figure 4.

git merge introduces a *merge commit* in the history of the feature branch, that binds the histories of both master and feature, and updates the feature with changes made to the master branch. Figure 3 shows the branching structure before *merge commit*. The branching structure after *merge commit* is shown in Figure 5.

Rebase: This has many more capabilities like rewriting the history interactively, re-sequencing the history, editing

```
(base) ekta@git-demo$ git branch
* feature
  master
(base) ekta@git-demo$ echo "Demonstrate rebasing" >> branch_file.txt
(base) ekta@git-demo$ git add branch_file.txt
(base) ekta@git-demo$ git commit -m "2nd commit on branch"
[feature cc3b2f8] 2nd commit on branch
 1 file changed, 1 insertion(+)
```

Figure 6: Changing some files in the *feature* branch to demonstrate rebasing

```
(base) ekta@git-demo$ git checkout master
Switched to branch 'master'
(base) ekta@git-demo$ echo "new file again" > file_2.txt
(base) ekta@git-demo$ git add file_2.txt
(base) ekta@git-demo$ git commit -m "this commit will be rebased in feature branch"
[master 3b8a9e2] this commit will be rebased in feature branch
 1 file changed, 1 insertion(+)
 create mode 100644 file_2.txt
```

Figure 7: Changing some files in the *master* branch to demonstrate rebasing

```
(base) ekta@git-demo$ git checkout feature
Switched to branch 'feature'
(base) ekta@git-demo$ git rebase master
First, rewinding head to replay your work on top of it...
Applying: first commit on a branch
Applying: 2nd commit on branch
(base) ekta@git-demo$ git log
commit 6494d1b27b4efbf49da78c3a294b8f4eac93ffff
Author: Ekta Nandwani <Ekta.Nandwani@littb.org>
Date:   Thu May 7 19:41:52 2020 +0530

 2nd commit on branch

commit 6494d1b27b4efbf49da78c3a294b8f4eac93ffff
Author: Ekta Nandwani <Ekta.Nandwani@littb.org>
Date:   Thu May 7 19:35:15 2020 +0530

  first commit on a branch

commit i9b300252c51f2ba521aa6fc6d6979498e891f42d (master)
```

Figure 8: Rebasing *master* onto *feature*

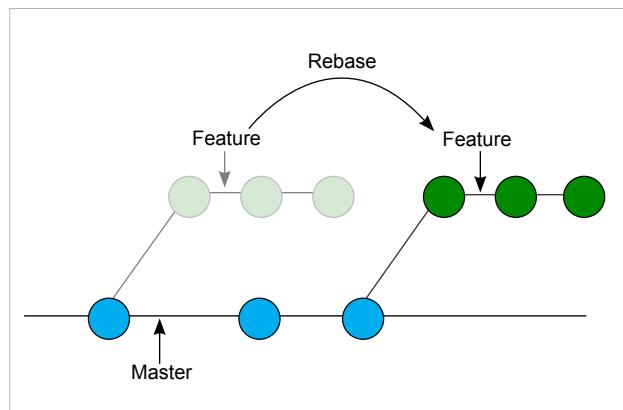


Figure 9: Branching structure after rebasing is done

```

(base) ekta@git-demo:~$ git remote -v
(base) ekta@git-demo:~$ git remote add origin https://github.com/ektanandwani/git-demo.git
(base) ekta@git-demo:~$ git remote -v
origin  https://github.com/ektanandwani/git-demo.git (fetch)
origin  https://github.com/ektanandwani/git-demo.git (push)
  
```

Figure 10: Adding a remote repository

commit messages, adding missing files in the commits, etc, but here we look at it in terms of integrating the work from two branches. This is called ‘rebasing source onto target branch’, which means taking the commit history from the source branch and reapplying target branch commits on top of it. It does not introduce *merge commit* unless there are conflicts with two branches. This leaves the target branch with a linear and clean history. This is done just like merging by checking out the target branch and rebasing the source onto the target. Some changes can be made in the *feature* and *master* branches to demonstrate rebasing, as shown in Figures 6 and 7.

```

(base) ekta@git-demo:~$ git push origin master
Username for 'https://github.com': ektanandwani
Password for 'https://ektanandwani@github.com':
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (15/15), 1.29 KB | 263.00 KiB/s, done.
Total 15 (delta 0), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/ektanandwani/git-demo.git
 * [new branch]      master -> master
  
```

Figure 11: Pushing the *master* branch to remote

```

(base) ekta@git-demo:~$ git pull origin master
From https://github.com/ektanandwani/git-demo
 * branch            master       -> FETCH_HEAD
Updating 19b3902..0c0da9e
Fast-forward
 README.md | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 README.md
  
```

Figure 12: Pulling changes from origin to *master* branch

Rebasing is done using *git rebase <source-branch-name>* as shown in Figure 8.

As seen in the log, the *feature* branch contains a full history of the *master* branch as well as its own commits applied on top of *master* branch commits. The new branching structure is shown in Figure 9.

Working with a remote repository

A remote repository is a local repository hosted somewhere on the Internet. It is generally used to collaborate with other people, be it the team members or anyone who has access to the repository. *git remote -v* is used to see the remote server URLs along with the configured shorthand for the same. There should be none by default in a newly initialised Git repository, but for a cloned repository, the

Branch	Commit Message	Date
williamtu Merge pull request #34 from antoninbas/add-2.13.0-release	Latest commit 9f831ac on 27 Feb	7 months ago
_data	Add list of talks for ovscon 2019.	7 months ago
_includes	Add 2.12.0 and older release announcements.	8 months ago
_layouts	Make the navbar mobile compatible.	3 years ago
assets	Add OVS logo in SVG format.	3 years ago
charter	Update links in charter docs to point to correct locations in source ...	3 years ago

Figure 13: A remote open source repository used for forking

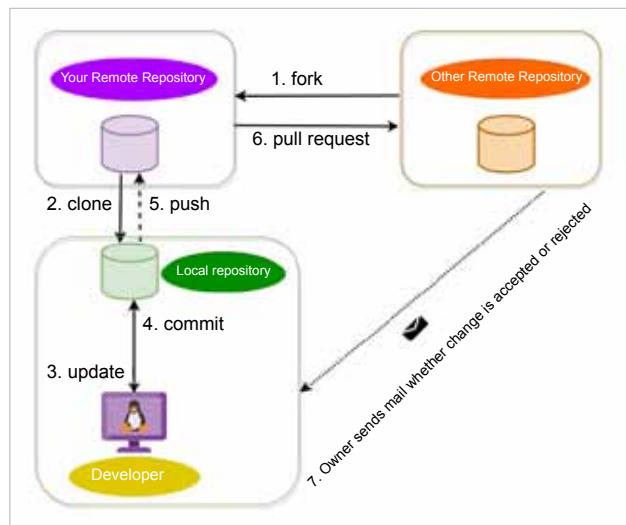


Figure 14: Block diagram demonstrating major commands

default would be the URL from which it was cloned with the shorthand as ‘origin’.

To add a new remote repository, the following command is used:

```
$ git remote add <shorthand> <url-to-remote-repo>
```

We have created a repository on GitHub to demonstrate working with remote repositories in Git, as shown in Figure 10.

Pushing the changes to the remote repository: The *git push* command is used to update the remote repository with the changes made in the local repository. It acts as a syncing function that uploads the local changes made in the codebase to the remote repository.

The basic command used is: *git push <remote> <branch>*. Here *<remote>* is the destination and *<branch>* is the source of the *push* command. An example of this could be: *git push origin master*. Here the *master* branch from the local repository is pushed to the origin (which is shorthand for the remote repository).

One thing to note is that *git push* only works when there is a fast-forwarding merge in the destination repository. A merge or any reference change in Git is called *fast-forward* if it is possible to traverse back to the old reference from the new reference. So, if the old reference is an ancestor of the new reference then it’s a *fast-forward*, which is explained in Figure 11.

Fetching the changes from the remote repository: The *git fetch* command downloads files, commits, refs, etc, from a remote repository to a local repository. The *fetch* command only downloads and doesn’t merge into the local repository. It has absolutely no effect on the local development. The head and the refs are not changed and no new *merge commit* is created. One can check out the *fetch*

content and see the changes in order to decide whether to merge it or not. The basic structure of the command is: *git fetch <remote>* where *<remote>* is the source remote repository.

Pulling the changes from the remote repository: The *git pull* command is used to pull in all the changes made in the remote repository to the local repository. The basic structure of the command is *git pull <remote> <branch-name>* which is shown in Figure 12.

The *git pull* command undergoes two internal operations: *git fetch* and *git merge*. *git fetch*, as described earlier, is used to download the contents of a branch from the specified remote repository. Then the *git merge* command is run, which merges the remote refs and heads into a new local *merge commit*. Instead of *merge*, *rebase* can also be done after *fetch*. For this, the *git pull --rebase <remote> <branch-name>* command is used.

Collaborating with open source projects

Every project/organisation has its own process with respect to collaboration. Forking a repository is one of the ways to collaborate with someone on a project or an open source project. For a demo, the OVS project at <https://github.com/openvswitch/openvswitch.github.io>, which is forked from GitHub, is shown in Figure 13.

Once the project is forked, it will be present in our GitHub account. The next steps are cloning it in a local machine, working on it like any other project and pushing the changes to the cloned repository. If we wish to contribute those changes back to the original repository, a *pull request* can be created with the proposed changes. The maintainer of the project will decide whether to accept or reject the proposed changes, and will send a notification regarding the same, as shown in Figure 14.

To summarise, Git provides developers a robust version control system which helps them in doing non-linear and parallel development. Because of its lightweight branches and consistent tracking, it provides scalability as well as immunity from data loss. These facts make Git an absolute favourite among developers. 

Reference

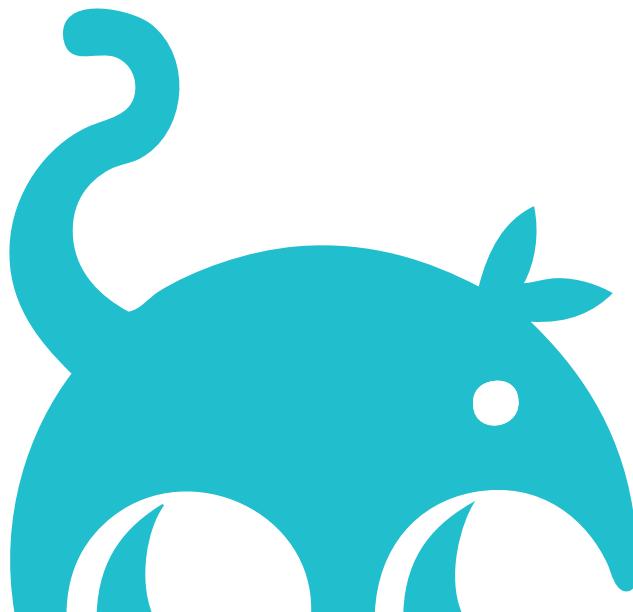
The entire Pro Git book, written by Scott Chacon and Ben Straub and published by Apress (<https://git-scm.com/book/en/v2>)

 By: Shubham Kumar, Ekta Nandwani and Prof. B. Thangaraju

The authors are associated with the Open Source Technology Lab at the International Institute of Information Technology, Bengaluru.

Use Sourcetrail to Work Easily with Unfamiliar Code

This article introduces the reader to Sourcetrail, a cross-platform source explorer, formerly known as Coati. Though Sourcetrail supports C, C++, Java and Python, we'll focus on C/C++ in this article.



 Have you ever tried to explore any source code without proper documentation? You will know the complexity of code analysis that is unfamiliar. Many cross-reference tools are available in the market, some of which are proprietary and some are limited to the command line. Sourcetrail is one such tool, which is free and open source, with a rich visual interface.

Source code analysis and visualisation support from this tool makes life easier for programmers, debuggers and testers alike, making them more productive with unfamiliar source code. Unlike debugging or profiling tools, this is a static analysis tool, i.e., it indexes the entire code

without execution. Thus, there is no narrowing down of indexing.

Setup

You can download Sourcetrail from <https://www.Sourcetrail.com/downloads> and follow the steps given in <https://www.Sourcetrail.com/documentation/#Installation> for installation. Here is a brief summary of the setup in the Linux environment:

- Download *Sourcetrail_xxx_Linux_64bit.tar.gz*
- Extract and run *install.sh* with admin privileges.
- This will install to */opt/Sourcetrail*, and *symlink* is created under */usr/bin/Sourcetrail*.
- You can now launch it using the

command *Sourcetrail* or create a desktop launcher shortcut.

Getting started

Once you launch Sourcetrail, a wizard appears to create a new project; you can open the existing project or choose from recent projects.

Let's go through some steps on how to create a new project and visualise some C++ source code as an example.

- Click on *Project -> New Project* and fill in the project name and location.
- Click on *Add Source Group*, and choose the *Empty C++ Source Group* under C++.
- Choose the desired C++ standard in the next wizard, and skip cross-compilation options for now.



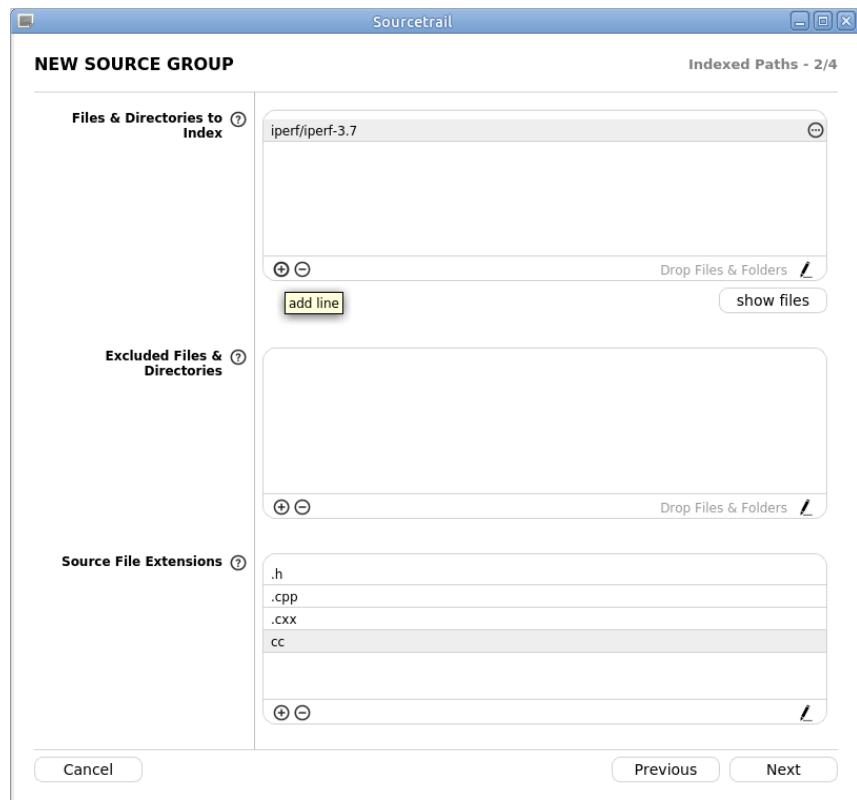


Figure 1: Attaching source group



Figure 2: Start indexing code

- Select the path in which you would like to index the source (files and directories to index).
- Choose *Source File Extensions* (typically, .h should be fine).
- If your source depends on external header files, please select *Include Paths*.
- Choose any compiler flags necessary, e.g., symbol definitions required for conditional inclusion.
- Finally, click on *Create*, and then on *Start*.

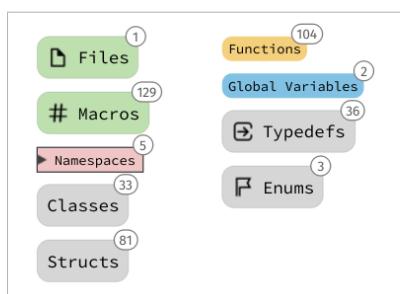


Figure 3: Code overview

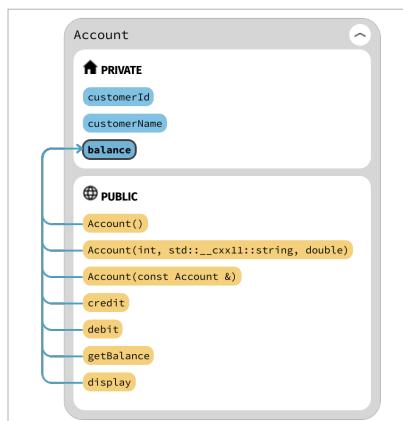


Figure 4: Visualisation of a class

A wizard will prompt you to start indexing code. The same wizard appears when you edit the project (menu path: *Project -> Edit Project*) or when you refresh the project manually.

Source visualisation and navigation

Once the indexing is complete, you can visualise the indexed code as external symbols like files, functions, classes, name spaces/packages, structs, unions, macros, typedefs, Enums, global variables, etc. The summary of indexing is on the right side pane.

You may ignore some errors initially, as the configuration is incomplete. To resolve these errors, you can update the configuration step by step, like the PATH of external header files, global include paths, compilation flags, filtering of file extensions, exclusion of some files and directories, and pre-compiled headers and flags. The indexing wizard appears on editing the project configuration (Figure 2).

You can navigate to any symbol under a particular section, which has a nice visualisation with suitable dependencies on the left side pane and concerned code on the right side. You can click on any cross-reference symbol, which will navigate to the concerned header/source file.

Figure 4 gives an example of a single class visualisation, with public and private members.

Figure 5 gives an example of visualising class relationships (inheritance in this case). For a better view, choose to show derived hierarchy/icons on the left side.

You can also view non-indexed symbols that are beyond the current scope of the code. At any point we can navigate between views using *Back* and *Forward* icons, list all past views with the *History* icon, and go back to the overview with the *Home* icon. You can use the *Refresh* icon when some files are updated externally and Sourcetrail cannot detect the changes; in this case,

the indexing wizard appears again (as in Figure 2). Equivalent menu options are available under the *History* menu.

Multiple tabs can be opened to have different views of the same code, e.g., classes in one tab, header files in another tab, etc.

We can take a snapshot of any view as an image, using the context menu option ‘Save as image’.

Attaching source code from other IDEs

Many times, you may find the source code under existing projects in popular IDEs like Visual Studio, Code Blocks, Qt Creator, etc. Examples are:

- Visual Studio solution for C/C++
 - Code Blocks project for C/C++
 - Maven/Gradle project for Java
- We can also attach source code in the form of clang JSON compilation database. You can generate this as follows for:
- CMake based projects, which define the flag *CMAKE_EXPORT_COMPILE_COMMANDS*.
 - Make projects; here, you can run the *make* command in the presence of the external tool Bear. You can build Bear from <https://github.com/rizsotto/Bear> or via package managers like apt in Ubuntu.
 - Qt Creator v4.8 (bundled with QtSDK 5.12) onwards comes with an option to generate a compilation database under the *build* menu.

Code editor plugins

Sourcetrail comes with a rich set of plugins to integrate with many popular IDEs/editors like Atom, Clion, Eclipse, Emacs, IntelliJ IDEA, Qt Creator, Sublime, Vim, VS Code and Visual Studio ID. With these plugins you can send the location from the editor to Sourcetrail, to see all the symbols found at that point. Some of these plugins support

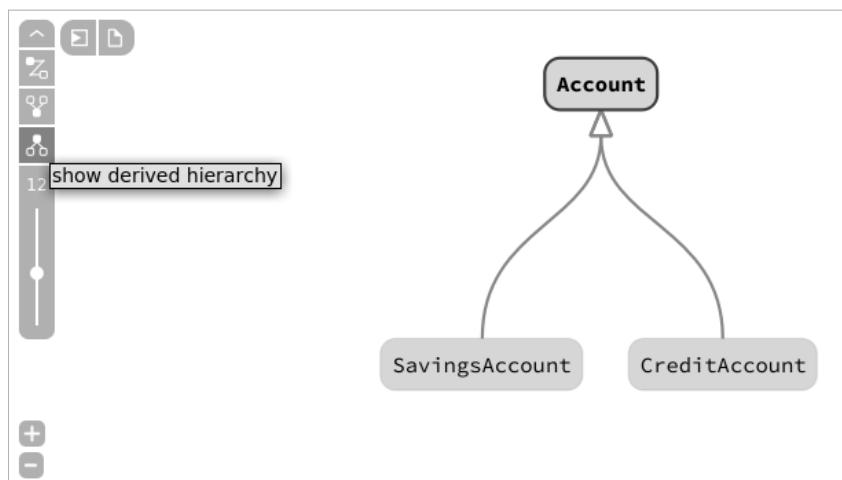


Figure 5: Relationships between classes

generation of the Clang compilation database, and you can attach this with Sourcetrail manually. Also, from the Sourcetrail code pane, you can switch to IDE, with the context menu option ‘Show in IDE’ for any symbol.

Please refer to <https://www.Sourcetrail.com/documentation/#CodeEditorPlugins> for the IDE/editor of your choice.

Source analysis fest

Here are some interesting libraries in C/C++; do try analysing a few of these with Sourcetrail, depending on your interest and the need to get a better understanding:

- JSON for modern C++: <https://github.com/nlohmann/json>
- iPerf API: <https://iperf.fr/iperf-download.php#source>
- QtSensors API: <https://github.com/qt/qtsensors>
- RTIMU library API: <https://github.com/RPi-Distro/RTIMULib>
- json-c API: <https://github.com/json-c/json-c>
- Point cloud library: <https://github.com/PointCloudLibrary/pcl>

com/PointCloudLibrary/pcl

- WiringPi API: <https://github.com/WiringPi/WiringPi>
- mbed driver, RTOS API: <https://github.com/ARMmbed/mbed-os/>
- curl APIs: <https://github.com/curl/curl>
- Paho C library: <https://github.com/eclipse/paho.mqtt.c>

Note: Some of these libraries have rich documentation; you can just compare the official documentation with the Sourcetrail outcome.

You can try some more libraries, as per your domain interest/background, in other supported languages too, such as Java and Python. You can also attach any custom code available with you and visualise it.

This article has given you a glimpse of the features of Sourcetrail, and some initial hands-on pointers. Please explore the tool in depth and gain confidence when dealing with unfamiliar code. **END**

By: Dr Prithvi Sekhar Pagala and Rajesh Sola

The authors are educators in the Education and Competency Development (ECoDe) team at KPIT Technologies.

OpenCV: An Excellent Tool for Computer Vision



Computer vision imparts human intelligence and instincts to a computer. This field of computer science works on enabling computers to see, identify and process images the same way the human eye does. OpenCV is a great tool to accelerate computer vision in commercial products.

Let us say you and your family went on a vacation and you uploaded a few pictures on Facebook. However, since it takes time to tag the names in each picture, Facebook is intelligent enough to do that for you. How do you think this auto-tag feature works? Well, it works through computer vision.

What is OpenCV?

OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. This library has more than 2500 algorithms used to detect and recognise human faces, identify images, track moving objects, and extract 3D models of objects.



Figure 2: New York's Times Square

Installation of OpenCV

To install OpenCV for Python, use the following code in the terminal:

```
$ python3 -m pip install opencv-python
$ python3 -m pip install opencv-contrib-python
```

How does a computer read an image?

One look at Figure 2, and we can see that it is a picture of the Times Square in New York. But computers cannot

analyse that, since they don't have any intelligence.

For any image, there are three primary colours — red, green and blue. A matrix is formed for every primary colour, and later, these matrices combine to provide a pixel value for the individual R, G, and B colours. Each element of these matrices provides data pertaining to the intensity of brightness of the pixel. It reads any image as a range of values between 0 and 255.

How are videos and images captured through a camera?

```
import cv2
cap = cv2.VideoCapture(0)
while True:
    ret, frame = cap.read()
    cv2.imshow("Capturing", frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

As seen in the above piece of code, we need to import the OpenCV library, `cv2.VideoCapture()` triggers the camera, and `cv2.imshow` shows what the camera is capturing by opening a new window. `cv2.waitKey` makes the window static until the user presses a key.

Basic functions of OpenCV

To load images using OpenCV and



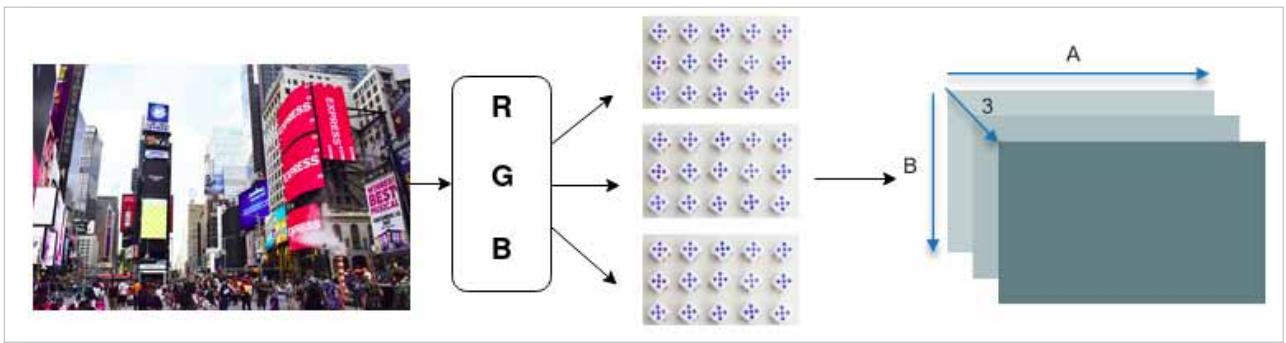


Figure 3: Conversion of an image to numpy arrays

converting them into grayscale, type:

```
import numpy as np
import cv2

img = cv2.imread('Image123.png',0)
#write the name of an image
cv2.imshow('image',img)
k = cv2.waitKey(0) & 0xFF
if k == 27:      # wait for ESC key to exit
    cv2.destroyAllWindows()
elif k == ord('s'): # wait for 's' key to save and exit
    cv2.imwrite('Firefox_wallpapergray.png',img)
    cv2.destroyAllWindows()
```

cv2.imread reads the selected image, and the *0* parameter turns the image into grayscale; *cv2.imshow* shows the converted image.

Drawing and writing on an image can be done as follows:

```
import numpy as np
import cv2
img = cv2.imread('black.jpg',cv2.IMREAD_COLOR)

cv2.line(i
mg,(0,0),(511,511),(255,0,0),5)
cv2.rectangle(i
mg,(384,0),(510,128),(0,255,0),3)
cv2.circle(img,(447,63), 63, (0,0,255),
-1)

font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(img,'OpenCV',(10,500),
```

font, 4, (255,255,255), 2, cv2.LINE_AA)

```
cv2.imshow('image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

cv2.line draws the line with the given coordinates on the image, *cv2.rectangle* and *cv2.circle* draw a rectangle and circle respectively, and *cv2.putText* writes the given text. Here, we have used the Hershey Simplex font. The output is shown in Figure 4.

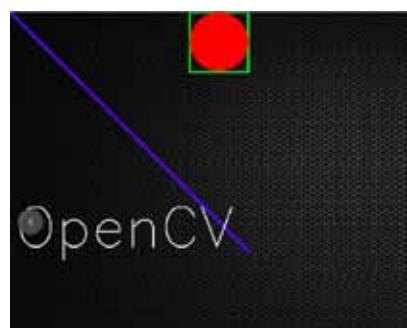


Figure 4: Output after drawing and writing on an image

Feature and template matching

Template matching is basically a part of one image matching another image. The code for this is as follows:

```
import cv2
import numpy as np
img_bgr=cv2.imread('sc1.png')
img_gray=cv2.cvtColor(img_bgr, cv2.COLOR_BGR2GRAY)

template=cv2.imread('sc2.png',0)
w,h=template.shape[::-1]
```

```
res=cv2.matchTemplate(img_
gray,template, cv2.TM_CCOEFF_NORMED)
threshold=0.8
loc=np.where(res>=threshold)
```

```
for pt in zip(*loc[::-1]):
    cv2.rectangle(img_bgr, pt,
(pt[0]+w, pt[1]+h), (0,255,255), 2)

cv2.imshow('detected',img_bgr)
cv2.waitKey()
```

cv2.imread reads the selected image, *cv2.cvtColor* converts the image into grayscale, *w* and *h* variables are the positions of x and y-axis, *cv2.matchTemplate* helps to match the common area of two images with threshold 80 per cent, and *cv2.rectangle* marks the area that is matching in the image.

The output is shown in Figures 5 and 6.

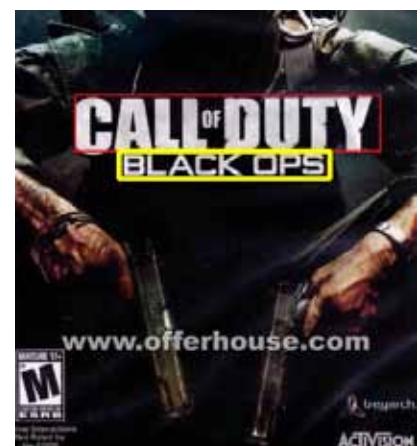


Figure 5: Output of template matching

BLACK OPS

Figure 6: The image we want to detect

We can see that the cropped image ‘black ops’ is matched with the portion of the full image, and the area is outlined by a yellow rectangle.

Gradients and edge detection

For detecting gradients and edge, type:

```
import cv2
import numpy as np
cap=cv2.VideoCapture(0)
while True:
    _, frame=cap.read()
    a= cv2.Laplacian(frame,cv2.CV_64F)
    x= cv2.Sobel(frame,cv2.CV_64F,1 ,0,
    ksize=5)
    y= cv2.Sobel(frame,cv2.CV_64F,0 ,1,
    ksize=5)
    edge= cv2.Canny(frame, 100, 200)
    cv2.imshow('original',frame)
    cv2.imshow('laplacian',a)
    cv2.imshow('sobelx',x)
    cv2.imshow('sobely',y)
    cv2.imshow('edge',edge)

    k=cv2.waitKey(5) & 0xFF
    if k == 27:
        break
cv2.destroyAllWindows()
cap.release()
```

`cv2.Laplacian` converts the image into the gradient, while `cv2.Sobel` converts it into a horizontal and vertical gradient. We use `cv2.CV_64F` as a standard label.

The output is shown in Figure 7.

Edge detection is pervasive in many applications such as fingerprint matching, medical diagnoses and licence plate detection. These applications basically highlight the areas where image intensity changes drastically, and ignore everything else.

Edge detection is also used in self-driving cars for lane detection.

Other features of OpenCV include motion detection, intrusion detection, homography, corner detection,



Figure 6: The image we want to detect



Figure 7: Laplacian



Figure 8: Vertical gradient



Figure 9: Horizontal Gradient



Figure 10: Edge detection

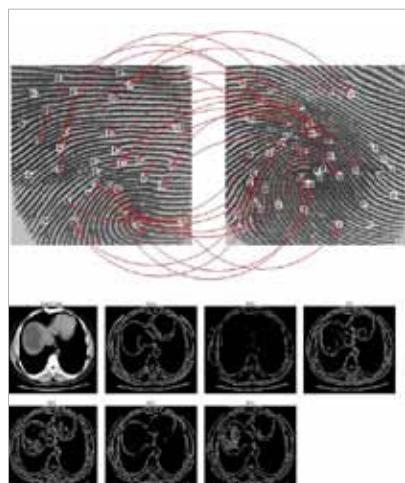


Figure 11: Fingerprint matching and edge detection of an embryo (Source: <https://www.quora.com/What-are-the-applications-of-edge-detection/answer/Ravindra-Sadaphule>)

colour filtering, thresholding, image arithmetic, etc.

Statistical machine learning libraries used by OpenCV include Naive Bayes classifier, K-nearest neighbour algorithm, decision-tree learning, meta algorithm, random forest, support vector machine, and deep and convolutional neural networks.



Figure 12: Lane detection of roads using OpenCV (Source: <https://www.youtube.com/watch?v=xdiekchp-Uc>)

Some popular applications of OpenCV are:

- Driver drowsiness detection (using a camera in a car) by alerting the car driver with a buzz or alarm.
 - Counting vehicles on highways (can be segregated into buses, cars, trucks) along with their speeds.
 - Anomaly detection in the manufacturing process (the odd defective product).
 - Automatic number plate recognition (ANPR) to trace vehicles and count the number of passengers.
- OpenCV is also used for data imaging to provide better diagnosis and treatment for a range of diseases. 

By: Aman Rangapur

The author is interested in the latest open source technologies and is an active contributor to the community.

Less is Actually More!

Less (which stands for Leaner Style Sheets) is a backwards-compatible language extension for CSS. Originally written in Ruby, it was later ported to JavaScript.

reated in 2009 by Alexis Sellier, Less or Leaner Style Sheets was inspired by Sass. It has a leaner feature set than Sass, and a syntax closely matching CSS, which Sass did not have at that time. In May 2012, Alexis turned over control and development of Less to a core team of contributors who now manage, fix and extend the language. The main difference between Less and other CSS pre-compilers is that the former allows real-time compilation via `less.js` by the browser.

Variables

As the name states, Less helps us to control commonly used values at a single place. We can see many values repeated in our CSS files very often. For example:

```
a,
.link {
  color: #428bca;
}

.widget {
  color: #fff;
  background: #428bca;
}
```

Variables make code easier to maintain by giving a way to control those values from a single location:

```
// Variables
@link-color:      #428bca;
@link-color-hover: darken(@link-color,
10%);

// Usage
a,
.link {
  color: @link-color;
}
a:hover {
```



```
color: @link-color-hover;
}

.widget {
  color: #fff;
  background: @link-color;
}
```

Mixins

Mixins are a way of including ('mixing in') a bunch of properties from one rule-set into another one. Say, we have the following class:

```
.bordered {
  border-top: dotted 1px black;
  border-bottom: solid 2px black;
}
```

Now suppose we want to use these properties inside other rule-sets. Well, we just have to drop in the name of the class where we want the properties, in this way:

```
#menu a {
  color: #111;
  .bordered();
}

.post a {
  color: red;
  .bordered();
}
```

The properties of the `.bordered` class will now appear in both `#menu a` and `.post a`. In normal CSS, we would have had to repeat this multiple times. As you can see, variables as well as mixins help us avoid

having repeated values in the file.

Nesting

Less gives you the ability to use nesting instead of, or in combination with, cascading. For example:

```
#header {
  color: black;
}

#header .navigation {
  font-size: 12px;
}

#header .logo {
  width: 300px;
}
```

In Less, we can also write it this way:

```
#header {
  color: black;
  .navigation {
    font-size: 12px;
  }
  .logo {
    width: 300px;
}
```

The resulting code is more concise and mimics the structure of your HTML at-rules. For example, `@media` or `@supports` can be nested in the same way as selectors. The at-rule is placed on top, and the relative order against other elements inside the same rule set remains unchanged. This is called bubbling.

```
.component {
  width: 300px;
  @media (min-width: 768px) {
    width: 600px;
    @media (min-resolution: 192dpi) {
      background-image: url(/img/
retina2x.png);
```



```

        }
    }
    @media (min-width: 1280px) {
        width: 800px;
    }
}

outputs:
.component {
    width: 300px;
}
@media (min-width: 768px) {
    .component {
        width: 600px;
    }
}
@media (min-width: 768px) and (min-resolution: 192dpi) {
    .component {
        background-image: url(/img/retina2x.png);
    }
}
@media (min-width: 1280px) {
    .component {
        width: 800px;
    }
}

```

Operations

Arithmetical operations such as +, -, *, /, can operate on any number, colour or variable. If it is possible, mathematical operations take units into account and convert numbers before adding, subtracting or comparing them. The result has the leftmost explicitly stated unit type. If the conversion is impossible or not meaningful, units are ignored. Examples of impossible conversion are px to cm or rad to %.

```

// numbers are converted into the same
units
@conversion-1: 5cm + 10mm; // result
is 6cm
@conversion-2: 2 - 3cm - 5mm; // result
is -1.5cm

// conversion is impossible
@incompatible-units: 2 + 5px - 3cm; //
result is 4px

```

```

        // example with variables
        @base: 5%;
        @filler: @base * 2; // result is 10%
        @other: @base + @filler; // result is 15%
    }
}
```

Multiplication and division do not convert numbers. It would not be meaningful in most cases — a length multiplied by a length gives an area, and CSS does not support specifying areas. Less operates on numbers as they are, and assigns explicitly stated unit type to the result:

```
@base: 2cm * 3mm; // result is 6cm
```

You can also do arithmetic on colours with Less:

```

@color: #224488 / 2; // results in
#112244
background-color: #112244 + #111; // 
result is #223355

```

Functions

Less provides a variety of functions that transform colours, manipulate strings and do maths. Using them is pretty straightforward. The following example uses percentage to convert 0.5 to 50 per cent, increases the saturation of a base colour by 5 per cent, and then sets the background colour to one that is lightened by 25 per cent and spun by 8 degrees:

```

@base: #f04615;
@width: 0.5;

.class {
    width: percentage(@width); // returns
    '50%'
    color: saturate(@base, 5%);
    background-color: spin(lighten(@base,
    25%), 8);
}

Maps
As of Less 3.5, you can also use mixins
and rulesets as maps of values.
#colors() {
    primary: blue;
    secondary: green;
}

```

```
.button {
    color: #colors[primary];
    border: 1px solid #colors[secondary];
}
```

As expected, this gives the output:

```
.button {
    color: blue;
    border: 1px solid green;
}
```

Sass and Less

Less is inspired by Sass. The latter was designed to both simplify and extend CSS, so things like curly braces were removed from the syntax. Less was designed to be as close to CSS as possible, and as a result existing CSS can be used as valid Less code.

To assign variables, Sass uses '\$' while Less uses '@'. The problem here is that there are a few existing CSS selectors that already use @. At times, this can be a little confusing. Sass and Less have extensions available to integrate mixins (the ability to store and share CSS declarations throughout a site). Sass uses Compass for mixins, which includes every option available with updates for future support. Less has Preboot, less, Less Mixins, Less Elements, gs, and Frameless. Software support for these extensions is supposedly quite fragmented, which means that we may have to use a combination of these.

Sass and Less were both initially on Ruby. Less was then ported to JavaScript. Much later, Ruby Sass was moved to Dart Sass, which compiles to pure JavaScript. 

References

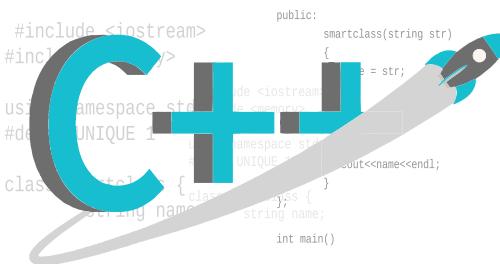
- [1] <http://lesscss.org/>
- [2] [https://en.wikipedia.org/wiki/Less_\(stylesheet_language\)#Less_software](https://en.wikipedia.org/wiki/Less_(stylesheet_language)#Less_software)

By: Athira Lekshmi

The author is an open source enthusiast.

Let's Get Familiar with Smart Pointers in C++

Many readers have probably heard about smart pointers and their benefits. One of the features of C++11, smart pointers are quite useful and have an inbuilt mechanism to ensure a program is free from memory leaks.



Before beginning, let's have a look at why people prefer smart pointers over normal pointers. When dealing with normal pointers, the onus is on programmers to allocate and release the resources or memory, in order to avoid memory leaks and exceptions. Programmers also need to prevent access to the resources that are already freed up, or it will lead to dangling references. That is why many beginners dislike normal or raw pointers.

Smart pointers and their types

Smart pointers are not just pointers; they are objects that are smarter. Unlike normal pointers, memory is released once the pointer goes out of scope.

unique_ptr: An object can be held by a pointer at a given time — multiple pointers cannot point to the same object. For example, consider a pointer *obj1* pointing to an object; another pointer named *obj2* cannot point to the object pointed by *obj1*. *unique_ptr* doesn't allow copying of pointers, but it allows transfer of ownership from one pointer to another.

- Allows exactly single ownership of the object.
- No two different pointers can hold the same object.

- Object cannot be shared or copied to another pointer but can be moved.
- It can manage a single object and dynamically allocate an array of objects.

```
#include <iostream>
#include <memory>

using namespace std;
#define UNIQUE 1

class smartclass {
    string name;
public:
    smartclass(string str)
    {
        name = str;
    }

    int display()
    {
        cout<<name<<endl;
    }
};

int main()
{
    unique_ptr<smartclass> P1(new
    smartclass("smartclass"));
}
```

```
P1->display();

#ifndef UNIQUE
    unique_ptr<smartclass> P2;
    P2 = move(P1);
#else
    unique_ptr<smartclass> P2(P1);
#endif

P2->display();

return 0;
}
```

The output is:

```
smartclass
smartclass
```

Comment `#define UNIQUE 1` in the above code and compile; then the compiler throws an error. *unique_ptr* doesn't have a copy constructor.

shared_ptr: Of all the smart pointers, this one is really smart. Multiple pointers can point to a single object. This is based on the shared ownership model. For example, pointers *obj1* and *obj2* can point to an object.

- It is based on the shared ownership model.
 - It has a reference counting mechanism; when each pointer gets released or goes out of scope the reference count decreases by 1.
 - When the reference count reaches 0, the resource is freed up. Reference count can be retrieved using the *use_count()* method.
- shared_ptr* manages two blocks:



- The control block (contains meta data)
- The object being managed

```
#include <iostream>
#include <memory>

using namespace std;
class smartclass {
    string name;
public:
    smartclass(string str)
    {
        name = str;
    }

    int display()
    {
        cout << name<<endl;
    }
};

int main()
{
    shared_ptr<smartclass> P1(new
smartclass("smartclass"));
    cout << "P1:";
    P1->display();

    shared_ptr<smartclass> P2;
    P2 = P1;

    cout << "P2:";
    P2->display();
    cout << "reference count
:"<<P1.use_count() << endl;

    shared_ptr<smartclass> P3(new
smartclass("myclass"));
    cout << "P3 :" ;
    P3->display();
    cout << "is p1 unique ? " <<
P1.unique()<<endl;
    cout << "is p3 unique ? " <<
P3.unique()<<endl;

    P3.swap(P2); // swapping of
pointers
    cout << "P3 after swapping :" ;
    P3->display();
}
```

```
return 0;
}
```

The output is:

```
P1 :smartclass
P2 :smartclass
reference count :2
P3 :myclass
is p1 unique ? 0
is p3 unique ? 1
P3 after swapping :smartclass
```

In the above program, P1 and P2 point to the same resource; hence, the reference count yields 2.

Then we check whether the pointer is unique or not, to find out whether the ownership is shared. For P1, ownership is shared, so *P1.unique()* returns 0. For P3, *P3.unique()* returns 1, which means its ownership is not shared with any other pointer. Finally, we swap P3 with P2 just to show that *shared_ptr* supports pointer swapping.

weak_ptr: Using *weak_ptr*, it is possible to access the object but not own it. To use *weak_ptr*, the object must be already owned by some other pointer (*shared_ptr*). For example, if a shared pointer *obj1* owns a resource, the same resource can be accessed by another *weak_ptr* without owning it.

- Used to avoid cyclic reference between *shared_ptr* references.
- It must be converted to *shared_ptr* in order to access the referenced object.

```
#include <iostream>
#include <memory>

using namespace std;
int main()
{

    weak_ptr<int> wptr;
    shared_ptr<int> sp = std::make_
shared<int>(10);
    wptr = sp;

    cout << "use_count = :"<<wptr.
```

```
use_count();
    shared_ptr<int> sptr = wptr.
lock();
    cout <<"\nvalue
:"<<*sptr<<endl;
}
```

The output is:

```
use_count = :1
value :10
```

In the above program, change the *weak_ptr* to *shared_ptr* and comment the last two lines of code; then compile and run, and the *use_count* value yields 2. On the contrary, if we use exactly the same code as above, the *use_count* won't increase as the *weak_ptr* doesn't take ownership.

- weak_ptr::lock():* Creates a *shared_ptr* to manage the object.
- weak_ptr::use_count():* Returns the number of *shared_ptr* objects.
- Std::make_shared:* Makes a single heap-allocation for both the control block and the managed object.

The aim of this article is to make each and every reader understand smart pointers. In today's world, everything is driven by software, directly or indirectly. But to make the software reliable and stable, it is imperative to understand concepts like smart pointers; otherwise, one has to spend many man-hours debugging and fixing issues. During my career, I have fixed many pointer related issues and noticed that many programmers don't have the patience to write code for releasing the resource. Smart pointers are very helpful in minimising bugs. Finally, coding is easy if we understand the purpose and usage of different methods. 

 By: Sree Venkatesh

The author is an engineer by profession, with 6+ years of experience in the field of embedded Linux development.

Open Source Testing Solutions that Make Software Reliable



This article lists a few open source testing tools that can help developers validate their software.

 With the arrival of the agile and DevOps practices, the software development industry is undergoing major disruption. The reliability and stability of software has become critical, increasing the importance of software QA. Organisations adopt DevOps practices and establish CI/CD pipelines to enable delivery of small chunks of software in fast and frequent iterations. QA teams are moving towards the ‘shift left’ approach, enabling early testing of the software. These test teams need to have a result-oriented collaboration with the development teams, who are also involved in the QA process. Development teams create and run unit tests, while the test teams validate these at the API and UI layers.

More and more application development is moving towards cloud native technologies. Applications are being built with service packages and containers, deployed as microservices, and managed on cloud elastic infrastructure through DevOps processes and continuous delivery workflows.

Test automation helps to achieve a larger test coverage and higher level of regression testing to ensure the stability of the software.

Performance engineering is replacing performance-testing techniques. The focus in the former is on analysing how to prevent and forecast software performance issues.

Open source testing tools

Test tools need to be selected depending on the type of applications

and interfaces they are exposed to. Their coverage should encompass all test areas, be it functional or non-functional testing. Functional test areas include Web UI, mobile app UI, and API validation. Non-functional test areas include performance, reliability, scalability and accessibility.

There is a range of open source tools that can help to validate the software in all these areas. Let’s take a quick look at some of these.

BDD based test automation framework testing tools

The BDD or Behaviour Driven Development framework is a software development approach that allows the tester or business analyst to create test cases in simple text language (English), so that even non-technical team members can understand what is going on in the software project. A few popular BDD based test frameworks are discussed below.



Cucumber: Cucumber (<https://cucumber.io/>) allows automated acceptance testing by executing the examples that optimally describe the behaviour of the application. It follows the Gherkin syntax form for creating scenarios. With more than 30 million downloads, it is the world's No. 1 tool for BDD adoption. It has cross-platform OS support and compatibility with most of the programming languages like Ruby, Java, Python and .NET.

Gauge: Gauge (<https://gauge.org/>) is an open source test automation framework released by ThoughtWorks, the creator of Selenium. It allows users to create readable and maintainable tests using the languages of your choice. It uses the Markdown syntax format for creating test specifications. Gauge has cross-platform and language support for writing test code. It supports Web, mobile and API interfaces for testing.

API automation testing tools

API testing is becoming critical since release cycles are becoming shorter with the adoption of agile and DevOps practices. With the 'shift left' testing approach, API testing bridges the gap between unit (application logic) and GUI layer testing.

Postman: Postman (<https://www.getpostman.com/>) is a good choice for API testing for those who don't want to deal with coding in an integrated development environment using the same language as the developers. Its characteristics are:

- Easy-to-use REST client
- Rich interface which makes it easy to use
- Can be used for both automated and exploratory testing
- Compactable on Mac, Windows, Linux and Chrome apps
- Has a bunch of integrations like support for Swagger and RAML formats
- Has run, test, document and monitoring features

Rest-Assured: Rest-Assured (<http://rest-assured.io/>) is an open source Java domain-specific language that makes testing REST services simpler. It has many inbuilt functionalities, which means users do not have to code things from scratch. It integrates seamlessly with the Serenity automation framework, so that users can combine the UI and REST tests in one framework to generate awesome reports.

Mobile test automation tools

Appium: Appium (<http://appium.io/>) can help to automate workflows for native, hybrid and Web apps. It works on the client-server architecture. Appium can automate workflows of iOS, Android and Windows apps. It provides support for multiple programming languages like Java, Ruby, Python, PHP, JavaScript and C#. The popularity of this mobile automation testing tool is attributable to its easy installation and usage.

Robotium: Robotium (<http://robotium.com/>) is an open source test automation framework used for Android UI testing. It supports gray box UI testing, system testing, functional testing and user acceptance testing for both native and hybrid Android based applications.

Katalon: Katalon (<https://www.katalon.com/>) is a free Web, API and mobile test automation solution. It is simple to use, provides cross-platform support, and has good JIRA integration. You can create tests, run tests, view reports and maintain tests very easily using Katalon studio. Katalon adoption has been rising in recent years, and it was picked as the customers' choice in a survey done by Gartner in 2019.

Performance testing tools

Apache JMeter: Apache JMeter (<http://jmeter.apache.org/>) is an open source Java desktop app, intended for load testing of Web applications. It has many good features like dynamic reporting, portability, powerful test IDEs, etc. It supports different types of

applications, protocols, shell scripts, Java objects and databases.

K6: K6 (<https://k6.io/>) is an open source load and performance testing tool for testing cloud native applications, APIs and microservices. It's a modern developer-centric CLI tool with test cases written in ES6 JavaScript, and with built-in support for HTTP/1.1, HTTP/2 and Web Socket protocols. K6 can easily be introduced into automation pipelines in Jenkins, GitLab, Azure DevOps, CircleCI and other CI/CD tools for performance regression testing.

Grinder: Grinder (<http://grinder.sourceforge.net/>) is an open source Java based load testing framework. It uses multiple load injector machines, which make running a distributed test quite easy. Its main features include a generic approach, flexible scripting, distributed framework and mature http support.

Reliability and stability testing tools

Netflix Simian Army: Simian Army (<https://github.com/Netflix/SimianArmy/wiki>) consists of services (monkeys) in the cloud for generating various kinds of failures, detecting abnormal conditions and testing the ability to survive them. The goal is to keep the cloud safe, secure and highly available. Currently, the simians include Chaos Monkey, Janitor Monkey and Conformity Monkey.

Litmus: Litmus (<https://github.com/litmuschaos/litmus>) is a toolset to do chaos engineering in a Kubernetes native way. Litmus provides chaos CRDs (custom resource definitions) for cloud native developers and site reliability engineers to inject, orchestrate and monitor chaos to find weaknesses in Kubernetes deployments.

Accessibility testing tools

As per World Bank estimates, around 15 per cent of the world population experiences some form of disability. Several countries have implemented legislation that makes it mandatory for organisations to ensure their products

and services are accessible to people with disabilities.

Accessibility testing ensures that the application is usable by people with disabilities like hearing and vision impairment, colour blindness and old age. Testing of the applications has to be compliant with accessibility standards like WCAG 2.1 level AA and AAA, Section 508, ADA and EN 301.

aXe: aXe (<https://www.deque.com/axe/>) is a free and open source accessibility testing tool by Deque Systems for Chrome and Firefox. It can be installed as an extension for Chrome or Firefox to analyse Web content. It provides a list of accessibility issues as part of its output. It also provides the exact piece of code that has caused the issue, along with the solution to fix it. It analyses the violations for WCAG 2.0 and Section 508 compliance.

Colour Contrast Analyser (CCA):

Colour Contrast Analyser (<https://github.com/ThePacielloGroup/CCAE>) helps to determine the legibility of text and the contrast of visual elements, such as graphical controls and visual indicators in your application.

It supports both Mac and Windows operating systems.

To keep up with the fast changing technology landscape, software QA organisations need to adopt newer techniques and tool sets. This article has given a brief description of the open source tools available for software testing. These can be used depending on the functionality and the application that needs to be tested. 

References

- [1] <https://www.digitalistmag.com/cio-knowledge/2019/02/26/evolution-of-technology-continues-what-is-next-in-2019-06196611>
- [2] <https://www.scnsoft.com/blog/software-testing-trends>
- [3] <https://dzone.com/articles/10-software-testing-trends-to-watch-out-for-in-201>
- [4] <https://www.softwaretestinghelp.com/open-source-testing-tools/>
- [5] <https://www.wipro.com/en-IN/consulting/accessibility/>

By: Srikanth Mohan

The author is a senior architect with 15+ years of experience in IT. He specialises in test automation and performance testing.

What's New In Electronics & IoT During These Challenging Times?

Visit 'Online Only' expo-cum-conference from the comfort of your HOME...



IndiaTechnologyWeek
@Home 2020

SEPTEMBER EDITION
16-17 SEP, 2020



VISIT NOW <https://IndiaTechnologyWeek.com>



CODE SPORT



In this month's column, we discuss a few coding questions.

In last month's column, we had discussed some of the recent state-of-art papers in Natural Language Processing from the Association of Computational Linguistics (ACL) conference. A few readers have asked me for pointers to follow up on recent research papers on different NLP topics. A good resource for this is the website <https://paperswithcode.com/>, which covers not just NLP but the whole of machine learning. For different NLP tasks, you can also browse through papers/techniques that achieve current state-of-art results on various benchmarks.

In this month's column, we take a break from our NLP discussion and instead discuss a few coding questions as practice problems for our readers. When you look at solving any coding problem, think about multiple ways of solving it. And for each of the approaches, think about the time and space complexity of the solution, and how you can optimise it. Remember that the solution's time and space complexity is often dictated by: (a) underlying data structure, and (b) the core underlying algorithm you are using. For instance, if the core underlying algorithm involves a linear scan/search through an array, you are going to end up with at least $O(n)$ time complexity, where 'n' is the size of the array. On the other hand, if your basic operation is a find-minimum on a binary heap, your complexity can reduce to $O(\log N)$. Even with a heap, if you need to search through every element, you will end up with a linear time complexity. So be careful about: (a) the data structures

you choose, and (b) the operations on the data structure you would be invoking. Let us look at an example problem to illustrate this.

You are given an array of unsorted distinct integers. You are asked to find the Kth largest element in the array. What is the time and space complexity of your solution?

The simplest approach is to sort the array in descending order. This incurs a time complexity of $O(\log N)$. And we can simply enumerate the element in the $(K-1)^{st}$ position in the sorted array as the Kth largest element (assuming array elements are indexed from zero). We don't need any additional space, since we can sort the array in place using a standard sorting technique. However, we are doing more work than what has been asked for in the problem, since now we can get all the elements in the right sorted position, not just the Kth largest. So how can we do less work to find just the Kth largest, which will help reduce the time complexity?

This, in turn, suggests that we need to look at techniques other than sorting. We know that we can get the maximum element from a max binary heap in $O(\log N)$. If we keep retrieving the maximum element K times, we get the Kth largest element in the Kth try. Given that it takes $O(N)$ to build a heap, and each extract-max takes $\log(N)$, our time complexity will be $O(k\log N)$. We don't need extra space since we can build the heap in place in the array itself. In this case, we retrieve the first largest element, the second largest element..., all the way up to the Kth largest element. But we are doing extra work again, because we were



asked to find only the Kth largest element. Are there other approaches we can consider to reduce this work?

Think about the quick sort technique, where we partition the array around a pivot element. After the partition step, the pivot element is in its correct and final position, while all the other elements are yet unsorted. But the catch here is that we can't find the pivot element in the Kth largest place. Therefore, we need a trial and error approach. We need to keep on trying pivot values such that we can hit upon the pivot element present at K-1st place in the final sorted array in descending order.

We can use the divide and conquer approach to reduce the work done in each step. Take a random element from the array, and partition the array around this element. Find the final resting index of your pivot element. Call it 'pivot_index'. If it is equal to (K-1), you have found the Kth largest element. If it is less than (K-1), you only need to consider the array starting from 'pivot_index + 1' for the Kth largest element. If the 'pivot index' is greater than (K-1), you need to consider only the array starting from 0 to 'pivot_index - 1'. Thus, you can end up reducing the work you do in each step, so that the array you need to examine for the Kth largest becomes smaller and smaller. In the average case, we can show that selecting the Kth largest element can be done in linear time. I will leave it to our readers to write the code for each of the above approaches as an exercise.

Now let us look at a few more coding questions that you can practice on.

1. There is a set of buildings on a street, standing from west to east. Any building that has a taller one on its west cannot have a view of the sunset. Can you find the buildings from which you can view the sunset? The building heights are given as a set of distinct integers in an array. What is the time and space complexity of your solution?
2. There is a set of N houses located on a street. The distance of each house from the start of the street is represented as a distinct integer and is provided in an array. You are asked to set up K mail boxes on the street, so that the total distance the residents from all houses need to walk to get their mails is minimised. Where would you place the K mail boxes? Let us think about some simplifying cases. If K is 1, where should you place the mail box? If K is N, where can you place the mail boxes? How can you generalise for any arbitrary value of K $1 \leq K \leq N$?
3. Consider an extension of the above problem. You are also given the number of residents for each house. This number can be varying widely for different houses, with values between 2 to 18. You are now asked to set up the K mail boxes such that the total

distance walked by all residents from all houses is minimised. Where would you place the K mail boxes? Again, solve this problem first for K = 1 and then for any arbitrary value of K.

4. You are given a terabyte sized file of IP addresses. You need to find an IP address that is not present in the file. Each IP address is a 32-bit integer. Note that fitting the entire set of addresses into memory is not possible. What is the time and space complexity of your solution?
5. You are given a stream of integers coming in. At every new integer coming in, you need to compute the median for the current set of integers seen so far. One possibility is to record all the numbers seen so far, and add the current newly incoming number to that set. And use this stored set of numbers to compute the median. This requires you to process all the elements at every step, and is wasteful. How can you come up with an incremental algorithm that does not redo the whole computation at each step?
6. Often, instead of being given a completely unsorted array, you are given an almost sorted array. Basically, each element has possibly been disturbed from its correct position by utmost K elements. This is known as a k-sorted array. For example, 1, 5, 2, 8, 9 is a 2-sorted array. Can you come up with a solution to get this array fully sorted? What is the time and space complexity?
7. You are given an array A of N integers. You need to create an array B from elements of A such that $B[0] \leq B[1] \geq B[2] \leq B[3] \geq B[4]$, and so on for N elements. Basically, the adjacent pairs of elements in B are sorted in opposite directions. How will you construct the array B from A? One possibility is to sort the entire array. Now divide the array into two halves, and interleave one element each from each half. Basically, you would end up with A[0], A[N/2+1], A[1], A[N/2+2], and so on. Alternatively, consider alternate pairs of the sorted array A and swap them — A[0], A[2], A[1], A[3], and so on. Both these approaches require O(nlogn) time complexity, since we end up sorting the entire array. Can you come up with a more efficient approach?

Feel free to reach out to me over LinkedIn/email if you need any help in your coding interview preparation. If you have any favourite programming questions/software topics that you would like to discuss on this forum, please send them to me, along with your solutions and feedback, at *sandyasm_AT_yahoo_DOT_com*. Wishing all our readers happy coding until next month! Stay healthy and stay safe. 

By: Sandya Mannarswamy

The author is an expert in natural language processing and is currently working as an independent researcher. Her interests include natural language processing, machine learning and AI.

Moodle Plugins for Online Education: A Quick Look at Interactive Content - H5P

Schools the world over are closed due to the COVID-19 pandemic. In such a scenario, methods have to be found to engage students in meaningful learning experiences. This series of articles will discuss Moodle, a popular free and open source learning management system (licensed under the GNU General Public License) used to run online courses. In fact, it is one of the most popular LMSs in the world.



The world is going through difficult times. Mankind has faced similar challenges time and again, overcome them, and often come up with new technologies that have not only solved the then existing problems, but have also made a long term impact on the planet. Consider the inventions made during World War 2 — nuclear energy, computers, rockets, etc. Today, we are being challenged by a global pandemic that has affected us in many different ways.

The educational sector is one of the worst hit. With schools and colleges closed for many months, online education has been taken up by many of these institutions. For a country like India, with tens of millions of children studying in schools and colleges, conducting online classes is a tough ask. There are technological, psychological and pedagogical challenges to overcome. Lack of Internet connectivity as well as the unavailability of televisions and smart devices are some of the tech challenges.

For many children, it is difficult to learn from video lectures alone. The isolated feeling of learning alone at home is a huge psychological challenge too. It is difficult to simulate a classroom environment when students are in different parts of the city or country.

From a teacher's viewpoint, what are the pedagogical challenges faced while running online courses? An excellent history teacher may not know how to record a video. An excellent Malayalam teacher might be a poor communicator in front of



the camera without his students. It is also difficult for teachers to measure the level of attentiveness of students while attending online classes. Yes! Conducting online classes is a huge challenge for teachers.

But luckily, technology offers us many solutions to counter this challenge. A relatively high Internet bandwidth in recent years, and software like video editing tools and Learning Management Systems (LMS) are some of the technological advances that are helping teachers conduct online classes more effectively. In this series of articles, we will introduce some Moodle plugins that will help educators conduct online classes far more effectively. Plugins add features and functionality to software. We believe online education is going to be more and more relevant in the post-COVID world. Some Moodle plugins are Interactive Content - H5P, BigBlueButtonBN, MooTyper, etc. We are sure the use of these plugins will make online education more interactive and effective.

Interactive Content - H5P is a plugin used to add H5P based interactive content to Moodle. This plugin is maintained by Frode Petterson, Thomas Marstrander, Pål Jørgensen and Svein-Tore Griff With. But before we begin to discuss the installation of this plugin, here's a bit more about Moodle. An article on two other Moodle plugins — CodeRunner and Virtual Programming Lab (VPL)—used to create online virtual programming labs has appeared earlier in OSFY. You can find it at <https://www.opensourceforu.com/2018/08/useful-moodle-plugins-for-programming-language-instructors/> in the OSFY portal.

Dashboard / Site administration / Plugins / Install plugins

Plugin installer

Install plugins from the Moodle plugins directory [?](#)

Install plugin from ZIP file [?](#)

ZIP package [!](#) [?](#)

Choose a file...
mod_hvp_moodle39_2020020500.zip

Accepted file types:
Archive (ZIP).zip

Show more...

Install plugin from the ZIP file

There are required fields in this form marked [!](#).

Figure 1: Installing plugins in Moodle

[com/2018/08/useful-moodle-plugins-for-programming-language-instructors/](https://www.opensourceforu.com/2018/08/useful-moodle-plugins-for-programming-language-instructors/) in the OSFY portal.

Let us first install Moodle locally in our system. We need a XAMPP server (a free and open source Web server consisting of Apache HTTP server, MariaDB database, and support for PHP and Perl programming languages) to do that. But an easy method is to use the Bitnami Moodle Stack installer consisting of Moodle and XAMPP provided by Bitnami, now owned by VMware. The installer file is about 190MB and the installation process is very smooth. All you have to do is click Next and provide an admin user name and a password when prompted. This way your systems can have Moodle locally.

However, please note that the Moodle copy installed in your system in this way is not accessible by the outside world. You have to create a Moodle server for this. But we make the assumption that your college or school has a Moodle server, and you have a role such as a teacher or course creator assigned to you, so that you can create courses and edit course content (Moodle has various roles like Admin, Teacher, Course Creator, Student, etc, with varying levels of privileges). But no need to worry if your institute hasn't thought about setting up a Moodle server for online courses, yet. You can try the features of all the plugins in the local copy of your Moodle,

First name / Surname ▲	Email address	Roles	Groups	Last access to course	Status
<input type="checkbox"/> —	—	—	—	—	—
<input type="checkbox"/>  Guru Guru	ggg@ggg123.com	Teacher 	No groups	1 min 57 secs	Active   
<input type="checkbox"/>  Shishya Shishya	sss@sss123.com	Student 	No groups	Never	Active   

Figure 2: Participant page of the course H5P101

and be ready when your institute set up a Moodle server. In fact, one of us (Manoj) tested all the plugin activities in his Moodle server, and the other (Deepu) tried the same on his local copy of Moodle and found that both work the same. Moreover, we urge you to do all these activities on your local copy while learning, so that no potential harm is done on your up and running Moodle server, if it exists.

Now that we have Moodle in our system, let us add two more users and assign them the roles of a teacher and a student, so that the teacher can add content to courses and add students to it. Once again, there is a warning! Please do all this in the local copy of your Moodle and not on the Moodle server maintained by your institute. Open a browser of your choice and type the URL, <http://127.0.0.1/moodle/login/index.php>. You will be taken to

Figure 3: Editing the contents of a course

Figure 4: Moodle activities and resources

the local copy of Moodle and prompted for your user name and password. As of now you only have a single admin user name and password provided during the installation. On entering this user name and password you will be taken to the dashboard of the local copy of Moodle. Now let us add a new user as a teacher to Moodle. Go to *Dashboard>Site administration>Users>Accounts>Add a new user* in Moodle. Let us call our teacher *Guru*. Choose the authentication method as *Manual accounts* and provide a password also. Notice that since we haven't set up an email server, this is the only possible method to create new users. Repeat the same process for adding a user named *Shishya*. Let us now try to create a course titled *H5P101* on Moodle. Go to *Dashboard>Site administration>Courses>Manage courses and categories>Add a new course* in Moodle and provide the course full name and short name as *H5P101*. Provide a course start date and end date, and choose the course format as *Weekly*. Now we have a course titled *H5P101* on the local copy of Moodle. Go to *Dashboard>Courses>H5P101>Participants* in Moodle, select the button *Enrol users* and add the user *Guru* as a teacher. Notice that the default role is that of a student. Students can be added later by the teacher. Our plan is to add H5P interactive content to this course by logging in as the teacher *Guru*. But before doing so, we need to install the Interactive Content - H5P plugin on Moodle.

First, download the zipped installation file of the Interactive Content - H5P plugin named *mod_hvp_moodle39_2020020500.zip* from https://moodle.org/plugins/mod_hvp. Now, go to *Dashboard>Site administration>Plugins>Install plugins* in Moodle. Figure 1 shows this window for installing plugins.

Now, drag and drop the zipped installation file of the Interactive Content - H5P plugin downloaded

earlier. Click on the button *Install plugin* from the ZIP file and you will be taken to the installation window of this plugin. In that window, click on the button *Continue* when prompted, and then the Interactive Content - H5P plugin will be successfully installed in your Moodle. Now log out from Moodle as the admin user, so that we can log in again as the teacher called *Guru* and add some H5P interactive content to the course H5P101. After logging in as *Guru*, add the user *Shishya* as a student to the course by going to *Dashboard>My courses>H5P101>Participants* in Moodle and clicking the button *Enrol users*. Figure 2 shows the participant page of the course H5P101.

Let us add some H5P interactive content to the course as the teacher *Guru*. Notice that a user with the privilege *Student* cannot add or edit any activity in a course. Go to *Dashboard>Courses>H5P101* in Moodle and click the button *Turn editing on*. Figure 3 shows the top of the course page for the course H5P101. You can see the button has changed to *Turn editing off*, and you

should remember to turn editing off later. Since we have used a weekly format for the course, we can see the course being divided into a number of weeks in Figure 3.

On the right side of the beginning block and all the blocks corresponding to each week of the course, you can see a link named '*Add an activity or resource*', which allows us to add activities and resources to this Moodle course. Let us add an H5P interactive activity in the beginning block itself. By pressing on this link named '*Add an activity or resource*', we will get the selection window shown in Figure 4, listing a lot of Moodle activities and resources. This figure shows us the number of options provided by Moodle while running an online course. We assure you that with such a rich set of activities and resources, you can run a highly interactive and informative online course. But in this article, we are interested in just one such activity, the H5P interactive content, marked with a bold green circle in Figure 4 to avoid any confusion.

By choosing the option *H5P Interactive Content*, you will be taken to a page with the window shown in Figure 5. This figure shows some of the interactive content offered by H5P. By scrolling down you can see a lot of H5P interactive content like Interactive Video, Course Presentation, Multiple Choice, Quiz (Question Set), Fill in the Blanks, Drag the Words, etc. This again tells us about the rich set of options provided by Moodle and H5P to create interactive and informative online courses. For want of space, we will demonstrate and discuss just one of these options, Quiz (Question Set), which allows different question types in the quiz. Click on the button *Get* shown in Figure 5, and you will see a window with a brief description, demo and an *Install* button. After installation you will see a button named *Use*, which can be clicked to use this particular option.

Notice that these steps need to be taken only once in the beginning for each option in the H5P interactive content list. From the second time onwards, we can simply click on the necessary option to include that activity. For this activity, we need to provide some mandatory information like *Title*, *Progress indicator* (whether to use a textual message or dots), *Pass percentage*, and at least one question. We named our quiz '*A Difficult Quiz*', and have chosen a textual progress indicator. It is possible to provide a background image, if required. We have given a sky blue image as the background for the quiz. We have set the pass percentage at 10. Now, we need to add some questions to this quiz. Figure 6 shows the default mode for setting questions, which tells us that the possible question types for a Quiz (Question Set) include Multiple Choice, Drag and Drop, Fill in the Blanks, Mark the Words, Drag Text and True/False Questions.

As an example, we will create a sample quiz with three questions —

The screenshot shows the 'Select content type' interface for H5P. At the top, there are two radio buttons: 'Create Content' (selected) and 'Upload'. Below them is a search bar labeled 'Search for Content Types' with a magnifying glass icon. The main area displays a list of 'All Content Types' (41 results). The items listed are:

- Multiple Choice**: Create flexible multiple choice questions. Includes a 'Get' button.
- Quiz (Question Set)**: Create a sequence of various question types. Includes a 'Get' button.
- Fill in the Blanks**: Create a task with missing words in a text. Includes a 'Get' button.
- Drag the Words**: Create text-based drag and drop tasks. Includes a 'Get' button.
- Column**: Organize H5P content into a column layout. Includes a 'Get' button.

At the bottom right of the list, there is a note: 'Activate Windows' followed by a 'Get' button.

Figure 5: Some of the interactive content of H5P

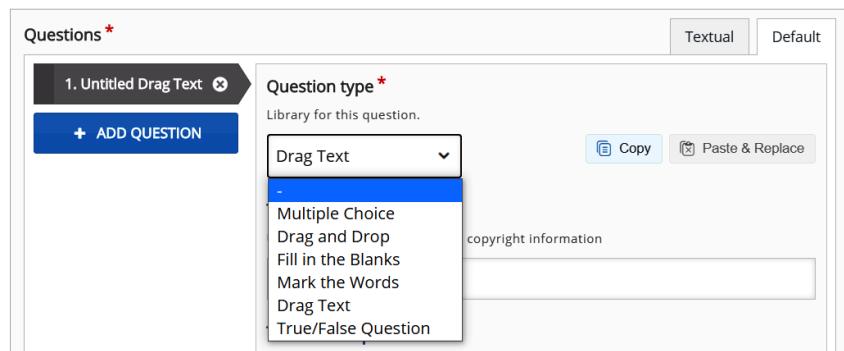


Figure 6: Question type options available for the Quiz (Question Set)

Figure 7: Setting up a fill in the blanks question

Figure 8: Setting up a true or false question

one fill in the blanks question, one true or false question, and one multiple choice question. First, let us add a fill in the blanks question. You have to choose the question type as *Fill in the Blanks*. A title is mandatory for each question. We have titled this question as ‘Do you know this proverb?’.

We are sure all of you have heard about

the famous proverb ‘A stitch in time saves nine’. The fill in the blanks question will look like the following: ‘A stitch in time saves ____’. You can set the question as shown in Figure 7. As you can see from the figure, the question is asked as follows, ‘A stitch in time saves *Nine/nine/NINE*’. Whatever comes within

the two asterisk (*) symbols will form the blank part in the question. If there are multiple correct answers, then they can be separated by a forward slash (/). In this example, ‘Nine’, ‘nine’ and ‘NINE’ are the three correct answers. Notice that, answers like, ‘9’, ‘nin’, ‘NinE’, ‘three’, ‘two’, etc, will all be treated as wrong answers.

Now let us add the second question, which is a true or false question.

To do so, click on the button **ADD QUESTION**. Now, choose the question type as *True/False Question*. We have given the title (which is mandatory) for this question as ‘Are you good at Mathematics?’.

Figure 8 shows setting up a true or false question. It is clear from this figure that the question is ‘ $3 + 3 = 7$ ’.

From Figure 8, it is also clear that you have to choose the correct answer for this question, either *True* or *False*. In this case, the answer is *False*, so that radio button is selected.

Now let us add a third and final question to this quiz — a multiple choice question. To do so, click on the button **ADD QUESTION** again. Now, choose the question type as *Multiple Choice*. We have given the title (which is mandatory) for this question as ‘A Tough Math Problem’.

Notice that it is slightly more complicated to set up a multiple choice question. Figure 9 shows the question as, ‘ $4 + 4 = ?$ ’ and one of the options for the answer is 8. Notice carefully that since this option is the correct response, it is ticked as *Correct*.

By default, two options can be added as an answer for a multiple choice question from the box while setting up such questions. But, if required, you can add more options as possible answers by clicking the button **ADD OPTION**.

We have added four options to this question. Now that we have added three questions to our example quiz, we can go down the page and click on any one of the two buttons to finalise the quiz — *Save and return to course* (which will take you to the course page) or *Save and display* (which will

Question *

$4 + 4 = ?$

Available options *

▼ 8

Text *

B **I** \times_2 \times^2 | T_x ≡ ≡ ≡

8

body div

Correct

Figure 9: Setting up a multiple choice question

take you to the quiz created). This quiz can be edited later, if required.

One great advantage with Moodle is the freedom to customise your tests. Even though we can proceed with the default settings while creating tests, a lot of freedom is offered to you so that you can conduct quizzes the way you like. For example, after a user finishes a test you can decide whether to show the result or not. Similarly, you can decide whether to show the solutions or not. You can also allow a user to reattempt questions and, if necessary, you can provide the user with immediate feedback after each attempt. It would be difficult to list out all the options available in

Moodle and the Interactive Content - H5P plugin, so that customised tests can be given to different groups of users. But with some patience, you will be able to do wonders.

Now that our teacher *Guru* has created an example test, let us logout and login again as the student, *Shishya*,

to try out the test. Login as *Shishya* and go to *Dashboard*>*My courses*>*H5P101* and you will see a link titled ‘A Difficult Quiz’ in the beginning block itself. Click on it and you will be taken to the first question. Notice that the questions are asked in a random order. So, if you take the test a second time, the order of the questions may change.

As mentioned earlier, there is a lot of interactive content that can be added by the Interactive Content - H5P plugin in Moodle. But before we wind up our discussion, we would like to reiterate the warning given by the development team of Interactive Content - H5P. Let us quote from the home page of Interactive Content - H5P, “Note that it is currently possible for people with programming skills to cheat in H5P interactions and obtain the full score without knowing the correct answers. H5Ps must not be used for exams or similar.” So, it is absolutely important not to use H5P for evaluation. You should only use it to make your classes interactive and make the learning process fun. If you want to carry out evaluation with Moodle, then there are a lot of time tested methods and activities to do so. 

By: Deepu Benson and Manoj Joseph

Deepu is a free software enthusiast whose area of interest is theoretical computer science. The open source tools of his choice include ns-2 and ns-3. The author maintains a technical blog at www.computingforbeginners.blogspot.in.

Manoj is currently working at Amal Jyothi College of Engineering, Kerala. His areas of interest are Big Data analytics and ICT. He is an expert in Moodle, and the open source tool he likes the most is Apache Hadoop.

What's New In Electronics & IoT During These Challenging Times?

VISIT NOW <https://IndiaTechnologyWeek.com>



Visit 'Online Only' expo-cum-conference
from the comfort of your HOME...



IndiaTechnologyWeek
@Home 2020

SEPTEMBER EDITION: 16-17 SEP, 2020

Demystifying Containers

This two-part series is all about understanding what happens behind the scenes when we run a `docker run <image>` command. Let us uncover the magic of container technology by building a container from scratch.

Ever since Docker released its first version back in 2013, it triggered a major shift in the way the software industry works. Lightweight VMs suddenly caught the attention of the world and opened opportunities for unlimited possibilities. Containers provided a way to get a grip on software that was built. Docker

containers could be used to wrap up an application in such a way that deployment and runtime issues like how to expose them on a network; how to manage their use of storage, memory and I/O; how to control access permissions, etc, were handled outside of the application itself, and in a way that was consistent across all containerised apps.

Containers offer many other benefits besides just handy encapsulation, isolation, portability and control. They are small in size (megabytes) and can start instantly. They have their own built-in mechanisms for versioning and component reuse. They can be easily shared via public or private repositories.

Today, containers are an essential component of the software development process. Many of us use them on a day-to-day basis. Despite this, there is still a lot of magic involved for many who want to venture into the world of containers in general. To date, there is a lot of ambiguity about how exactly a container works. The two articles in this series will try to demystify the working of containers. But before that, I believe we must understand how containers came to be.



The world before containers

For many years now, enterprise software has typically been deployed either on bare metal (i.e., installed on an operating system that has complete control over the underlying hardware) or in a virtual machine (i.e., installed on an operating system that shares the underlying hardware with other guest operating systems). Naturally, installing on bare metal made the software painfully difficult to move around and difficult to update — two constraints that made it hard for IT teams to respond nimbly to changes in business needs.

Then virtualisation came along. Virtualisation platforms (also known as hypervisors) allowed multiple virtual machines to share a single



physical system — each virtual machine emulating the behaviour of an entire system, complete with its own operating system, storage and I/O, in an isolated fashion. IT infra could now respond more effectively to changes in business requirements, because VMs could be cloned, copied, migrated, and spun up or down to meet demand or conserve resources.

Virtual machines also helped cut costs, because more VMs could be consolidated onto fewer physical machines. Legacy systems running older applications could be turned into VMs and physically decommissioned to save even more money.

But virtual machines still have their share of problems. They are large in size (gigabytes), and each one contains a full operating system. Only some virtualised apps can be consolidated onto a single system. Provisioning a VM still takes a fair amount of time. Finally, the portability of VMs is limited. After a certain point, VMs are not able to deliver the kind of speed, agility and savings that fast-moving businesses are demanding these days.

Containers

Containers work a little like VMs, but in a far more specific and granular way. They isolate a single application and its dependencies — all of the external software libraries the app requires to run — both from the underlying operating system and from other containers. All of the containerised apps share a single, common operating system, but they are compartmentalised from one another and from the system at large.

Taking an example of Docker, if you run `docker run -it ubuntu bash` in your terminal, you can see that like your host OS, your container too, has its own hostname and its own set of processes

What this means is that your Ubuntu container is running in an isolated environment. PID 1 confirms this fact. Similarly, we can provide a mounted storage to our container,

or allocate a particular number of processes or a certain amount of RAM to run with. But what exactly is all this? What exactly is process isolation? What is a containerised environment? What do metered resources mean?

We will try to make sense of all this jargon. We will try to replicate the behaviour of `docker run <image>` as close as possible. To make it all happen, we will be using Go as a language. There is no specific reason behind the selection of Go, though. You can choose any language like Rust, Python, Node, etc, too. The only requirement is that the language should support syscalls and namespaces. I picked Go because I have a personal preference for it. The fact that Docker is built on Go also helps my case.

Building a container from scratch

As mentioned earlier, we will try to replicate something as close to Docker as possible. Just like `docker run <image> cmd args`, we will go for `go run main.go cmd args`. To start with, we will proceed with the basic snippet that most Go plugins of all the major editors have to offer.

```
package main

func main() {
}
```

Now we will add support for the execution of basic commands like `echo` and `cat`:

```
func must(err error) {
    // If error exists, panic and exit
    if err != nil {
        panic(err)
    }
}
```

```
func run() {
    fmt.Printf("Running %v\n",
os.Args[2:])

    // Execute the commands that follow
'go run main.go run'
cmd := exec.Command(os.Args[2],
os.Args[3:]...)
cmd.Stdin = os.Stdin
cmd.Stdout = os.Stdout
cmd.Stderr = os.Stderr

must(cmd.Run())
}

func main() {
    // Make sure that the first argument
after 'go run main.go' is 'run'
switch os.Args[1] {
case "run":
    run()
default:
    panic("I'm sorry, what?")
}
}
```

Figure 1 shows what that boils down to.

Now that we can run simple commands with our script, as can be confirmed in Figure 1, we will now try running a bash shell. Since it can get confusing as we are already in a shell, we will try to run `ps` before and after running our script.

Based on Figure 2, it is still difficult to say anything. To confirm if we have isolation like an actual container, let us simply change the hostname from within our bash shell launched using our script. To modify or make any change in the hostname, we need to be a *root*:

The results in Figure 3 mean that we do not have isolation yet. To address this, we need the help of namespaces.

```
lubuntu@lubuntu:~/Projects/make-sense-of-containers$ go run main.go run echo "Hello, world!"
Running [echo Hello, world!]
Hello, world!
lubuntu@lubuntu:~/Projects/make-sense-of-containers$
```

Figure 1: Executing the `echo` command in our container

```

lubuntu@lubuntu:~/Projects/make-sense-of-containers$ ps
 PID TTY      TIME CMD
 1999 pts/0    00:00:00 bash
 2837 pts/0    00:00:00 ps
lubuntu@lubuntu:~/Projects/make-sense-of-containers$ go run main.go run bash
Running [bash]
lubuntu@lubuntu:~/Projects/make-sense-of-containers$ ps
 PID TTY      TIME CMD
 1999 pts/0    00:00:00 bash
 2838 pts/0    00:00:00 go
 2871 pts/0    00:00:00 main
 2875 pts/0    00:00:00 bash
 2883 pts/0    00:00:00 ps
lubuntu@lubuntu:~/Projects/make-sense-of-containers$ 

```

Figure 2: Observing the output of `ps` in the host and from within our container

```

root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# ps
 PID TTY      TIME CMD
 2930 pts/0    00:00:00 sudo
 2931 pts/0    00:00:00 bash
 3048 pts/0    00:00:00 ps
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# hostname
lubuntu
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# go run main.go run bash
Running [bash]
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# ps
 PID TTY      TIME CMD
 2930 pts/0    00:00:00 sudo
 2931 pts/0    00:00:00 bash
 3050 pts/0    00:00:00 go
 3081 pts/0    00:00:00 main
 3090 pts/0    00:00:00 bash
 3097 pts/0    00:00:00 ps
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# hostname
lubuntu
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# hostname pratikms
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# hostname
pratikms
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# exit
exit
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# hostname
pratikms
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# 

```

Figure 3: Attempting to change hostname from the container and verifying it later in the host

Namespaces

Namespaces provide the isolation needed to run multiple containers on one machine while giving each what appears like its own environment. There are six namespaces, and each can be independently requested.

PID: The PID namespace gives a process and its children their own view of a subset of the processes in the system. This is analogous to a mapping table. When a process of a PID namespace asks the kernel for a list of processes, the kernel looks at the mapping table. If the process exists in the table, the mapped ID is used instead of the real ID. If it doesn't exist in the mapping table, the kernel pretends it doesn't exist at all. The PID namespace makes the first process created within it PID 1 (by mapping whatever its host ID is to 1), giving the appearance of an isolated process tree in the container. This is a really interesting concept.

MNT: In a way, this one is the most important. The mount namespace gives the processes contained within it their own mount table. This means they can mount and unmount directories without affecting other namespaces including the host namespace. More importantly, in combination with the *pivot_root syscall*, it allows a process to have its own file system. This is how we can have a process think it's running on Ubuntu, CentOS, Alpine, etc — by swapping out the file system that the container sees.

NET: The network namespace gives the processes that use it their own network stack. In general, only the main network namespace (the one that the processes start when you start your computer use) will have any real physical network cards attached. But we can create virtual Ethernet pairs — linked Ethernet cards where one end can be placed in one network namespace and one in another, creating

a virtual link between the network namespaces. This is like having multiple IP stacks talking to each other on one host. With a bit of routing magic, this allows each container to talk to the real world while isolating each to its own network stack.

UTS: The UTS namespace gives its processes their own view of the system's hostname and domain name. After entering a UTS namespace, setting the hostname or the domain name will not affect other processes.

IPS: The IPC namespace isolates various inter-process communication mechanisms such as message queues. This particular namespace deserves a blog post of its own. There's so much to IPC than what I can comprehend myself. This is why I will encourage you to check out the namespace docs (<https://www.man7.org/linux/man-pages/man7-namespaces.7.html>) for more details.

USER: The user namespace was the most recently added and is likely the most powerful from a security perspective. This namespace maps the UIDs to different sets of UIDs (and GIDs) on the host, which is extremely useful. Using a user namespace we can map the container's root user ID (i.e., 0) to an arbitrary and unprivileged UID on the host. This means we can let a container think it has root access without actually giving it any privileges in the root namespace. The container is free to run processes as *uid 0* — which normally would be synonymous with having root permissions, but the kernel is actually mapping that UID under the covers to an unprivileged real UID belonging to the host OS.

Most container technologies place a user's process into all of the above namespaces and initialise them to provide a standard environment. This amounts to, for example, creating an initial Internet card in the isolated network namespace of the container, with connectivity to a real network on the host. In our case, for satisfying our immediate requirement, we will add the UTS namespace to our script

so that we can modify the hostname.

```
func run() {
    // Stuff that we previously went
    over

    cmd.SysProcAttr = &syscall.
    SysProcAttr{
        Cloneflags: syscall.CLONE_NEWUTS,
    }

    must(cmd.Run())
}
```

Running it returns what is shown in Figure 4.

Awesome! As per Figure 4, we now can modify the hostname in our container-like environment without letting the host environment change.

But if we observe closely, our process IDs within the container are still the same. We're able to see the processes running in our host OS even from within our container.

To fix this, we need to use the PID namespace. As discussed above, the PID namespace will allow us process isolation.

```
func run() {
    // Stuff that we previously went
    over

    cmd.SysProcAttr = &syscall.
    SysProcAttr{
        Cloneflags: syscall.CLONE_NEWUTS
        | syscall.CLONE_NEWPID,
    }

    must(cmd.Run())
}
```

However, unlike the case of UTS namespace, simply adding the PID namespace here like this won't help. We will have to create another copy of our process so that it can be run with PID 1:

```
func run() {
    cmd := exec.Command("/proc/self/
    exe", append([]string{"child"},
```

```
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# ps
  PID TTY      TIME CMD
 2930 pts/0    00:00:00 sudo
 2931 pts/0    00:00:00 bash
 3254 pts/0    00:00:00 ps
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# hostname
lubuntu
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# go run main.go run bash
Running [bash]
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# ps
  PID TTY      TIME CMD
 2930 pts/0    00:00:00 sudo
 2931 pts/0    00:00:00 bash
 3256 pts/0    00:00:00 go
 3289 pts/0    00:00:00 main
 3296 pts/0    00:00:00 bash
 3303 pts/0    00:00:00 ps
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# hostname
lubuntu
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# hostname pratikms
pratikms
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# hostname
pratikms
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# exit
exit
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers# hostname
lubuntu
root@lubuntu:/home/lubuntu/Projects/make-sense-of-containers#
```

Figure 4: Modifying hostname in our container and confirming that host OS is not affected

```
os.Args[2:]...)...)
cmd.Stdin = os.Stdin
cmd.Stdout = os.Stdout
cmd.Stderr = os.Stderr

cmd.SysProcAttr = &syscall.
SysProcAttr{
    Cloneflags: syscall.CLONE_NEWUTS
    | syscall.CLONE_NEWPID,
}

must(cmd.Run())

func child() {
    fmt.Printf("Running %v as PID
%#d\n", os.Args[2:], os.Getpid())

    cmd := exec.Command(os.Args[2],
os.Args[3:]...)
    cmd.Stdin = os.Stdin
    cmd.Stdout = os.Stdout
    cmd.Stderr = os.Stderr

    must(cmd.Run())
}

func main() {
switch os.Args[1] {
case "run":
    run()
case "child":
    child()
default:
```

```
panic("I'm sorry, what?")
}
}
```

What we are doing is that whenever we run `go run main.go run bash`, our `main()` function will be called. As the value of `os.Args[1]` will be 'run' at this instance, it will call our `run()` function. Within `run()`, we are using `/proc/self/exe` to create a copy of our current process. We are essentially creating a copy and calling it again by appending the string 'child' to it, followed by the rest of the arguments that we received in `run()`. When we do this, our `main()` function will be invoked again, with the difference being that the value of `os.Args[1]` will be 'child' this time. From there on, the rest of the script executes as we saw before.

But, even after all this, do we really have process isolation? Do we have resource limiting? Answers to that and more will be covered in the next part of this series. Watch out for this space! END 

 By: Pratik Shivaraikar

The author is an open source software enthusiast, evangelist, and contributor with varied experience of working in the storage, security, and wireless domains. He actively works on Go, Python, Node.js, PHP, and more. He writes at <https://blog.pratikms.com>.

Emergence of Pen Testing in Cyber Security

A penetration test is basically a simulated cyber attack against the system in order to check it for vulnerabilities. It is also called 'ethical hacking' or 'white hat hacking'. Pen tests are essential to assess the current security of an IT system. They provide valuable information about security risks and vulnerabilities, and their potential impact on the system's functioning and performance.



Image source: <https://indiancyberinstitute.com>

The cyber threat landscape is changing on a daily basis. Even as one threat is discovered, many unknown ones are waiting in the wings. However, business and corporations can today defend themselves through pen testing. A pen test includes a set of procedures used to discover any unknown weaknesses in the network infrastructure of a business.

Pen testing was first thought of in the 1960s. In 1971, the US Air

Force ordered security testing of its time shared computer systems. As hacking became more advanced, so did penetration testing.

After the WannaCry ransomware attack in May 2017, the concept of pen testing slowly started catching the attention of cyber security personnel. That attack locked more than 200,000 computers around the world, and demanded ransom payments through Bitcoin cryptocurrency. It affected many big organisations across the globe.

Why pen testing?

Pen testing can prove critical in the prevention of cyber attacks. Over 69 per cent of the organisations based in the US do not think their anti-virus protection or firewalls can effectively protect them from cyber attacks. And in 2017, a cyber attack was recorded every 40 seconds in the world, which resulted in total losses of US\$ 5 billion, an astonishing increase from the US\$ 325 million loss in 2015.



Penetration testing is mainly required for:

- Transferring financial or such critical data between different systems or over the network.
- To protect user data.
- To detect security weaknesses in an application.
- To discover defects in the system.
- To assess the losses due to cyber attacks and their impact on business.
- To meet the information security standards in the organisation.
- To bring into action an effective network security system in the organisation.

According to Hacken, the average cost of a penetration test can vary from US\$ 4,000 to US\$ 100,000.

By using various methodologies to prevent advanced attacks, pen testing can identify SQL injections, cross-site scripting, low frequency interface (LFI), radio frequency interface (RFI) and other defects in the organisation's Web applications and infrastructure.

Companies that carry out penetration testing

Automated tools can be used to identify some standard threats present in an application. Pen test tools can scan code to check if there is a malicious code present that can lead to potential security threats.

Science Soft is a recognised IT consulting and software development pioneer in cyber security services. The company has more than 16 years of experience in conducting black box, white box and grey box penetration testing of all the IT infrastructure components.

Geneva based software company ImmuniWeb has DevSecOps-enabled application penetration testing platforms. It offers comprehensive application penetration testing of internal and external Web and mobile apps, APIs and Web services, mail servers, IoT devices, and more.

Types of pen testing

Social engineering test: This test is done to protect personal sensitive information like passwords, business critical data, etc. This is done for phone, Internet or helpdesk processes. Sensitive information may be leaked in email or phone communications. Cyber hackers may take the benefit of this. Security audits can be conducted to identify and correct such process defects.

Web application test: This is concerned with testing Web applications.

Physical penetration test: This test is generally used in military and government facilities. All physical network devices and access points are tested for any security breach.

Network services test: This is concerned with testing services offered by a network.

Client side test: This tests client side servers, computers, laptops, and various other devices.

Remote dial-up test: This searches for modems in the environment, and tries to log into the systems connected through these modems by password guessing or brute-forcing.

Wireless security test: This is concerned with detecting open, unauthorised and less secure hotspots or Wi-Fi networks.

Penetration testing comprises three parts.

Black box penetration testing: In this methodology, the tester assesses the target system, network or process without the knowledge of its details. No code is examined in this method.

White box penetration testing: In this approach, the tester is equipped with complete details about the target environment — systems, network, OS, IP address, source code, schemes, etc. It examines the code and finds out design and development errors. It is a simulation of an internal security attack.

Grey box penetration testing: Here, the tester has limited details about the target environment. It is a simulation of external security attacks.

Tools used in pen testing

Given below are the top tools used by pen testing teams worldwide.

The Network Mapper (also known as NMAP): This free tool is used mainly for discovering any kind of weakness or holes in the network environment of a system. Here, you can create a virtual map of the network segment, and then detect major areas of weaknesses that cyber attackers can penetrate through without any difficulty. It comes in both command and GUI formats.

Metasploit: This is a package of different pen testing tools. It is powered by the Perl platform and comes with built-in customisable exploits that can be used to execute any kind of pen test.

Wireshark: This tool can analyse actual network protocols and data packets. You can detect any weakness in the network in real-time. Here, live information and data can be collected from Bluetooth, Token ring, Kerberos, WEP, any Ethernet based connections, etc.

John The Ripper (JTR): One of the biggest cyber security threats is the inherent weakness of the traditional password. JTR is mainly used to provide database security. It can be used to pen test password databases that are in online and offline mode.

Kali Linux: This advanced pen testing software is available only on Linux machines. Many experts believe that this is the best tool for both injection and password sniffing. Kali is optimised every way for a penetration tester. Open testers commonly use Kali virtual machines on OSX or Windows.

Hashcat: This is the world's fastest and most advanced password recovery utility tool. It detects many kinds of password-guessing based brute force attacks, including dictionary and mask attacks.

Forty-nine years after ‘creeper system’, a computer based self-replicating virus was created, the cyber security world is yet to win the cyber war. According to Cyber Risk Analytics, the first half of 2019 saw more than 3,800 publicly disclosed breaches with more than 4.1 billion records exposed.

A software exporter has hired a forensic firm to investigate cyber attacks on its network. It is in the process of building a new private email network because the intruders were believed to have compromised its corporate email system. Wipro says it investigates over 4.5 million security alerts a year, and puts employees through cyber security training courses.

Cyber security systems today won't be the same as a few weeks from now. This is especially true for organisations that maintain and develop software. As the threat landscape changes, it is important

for organisations to regularly conduct penetration testing on their critical assets. Pen testers can be called cyber heroes, as they go about their job of trying to achieve complete cyber security. **END** 

References

- [1] <https://www.softwaretestinghelp.com>
- [2] <https://testsources.infosecinstitute.com>
- [3] <https://www.csonline.com>
- [4] <https://www.alpinesecurity.com>blog>
- [5] 'Wipro hires forensic firm to probe cyber attack', an article by J. Mendonca that appeared in economictimes.com

By: Vinayak Ramachandra Adkoli

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



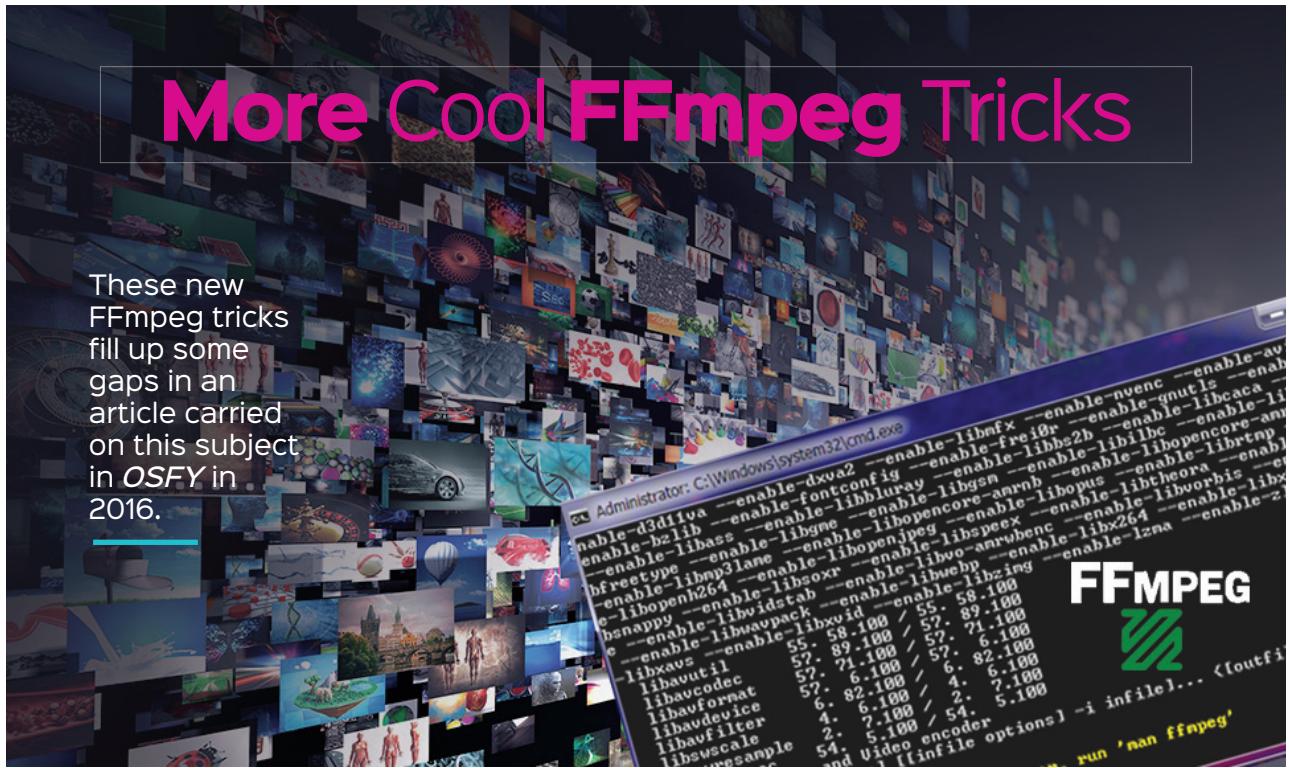
THE COMPLETE MAGAZINE ON OPEN SOURCE



- To find dealers to buy copy from news-stand, visit: www.ezine.efymag.com
- To subscribe, visit: pay.efyindia.com
- To buy an ezine edition, visit: www.magzter.com choose Open Source For You

More Cool FFmpeg Tricks

These new FFmpeg tricks fill up some gaps in an article carried on this subject in *OSFY* in 2016.



This is a follow-up to the article ‘Cool FFmpeg Tricks’ (<https://www.opensourceforu.com/2016/05/cool-ffmpeg-tricks/>) I wrote for the readers of *OSFY* in 2016. Early this year, I found that the number of FFmpeg-related options in my Nautilus

context menu had increased so much that there was enough material to write another FFmpeg article.

Then, I thought, why not combine the two and write a book on FFmpeg? I then started on a self-publishing spree that lasted until I had six books to my name. (My book publishing process is entirely powered by FOSS — Eclipse, MarkDown/ CommonMark, KHTMLToPDF, GIMP, Inkscape, Itext, ImageMagick and Calibre. I did not spend a paisa on anything other than Internet and electricity.) You can find my books at <https://www.amazon.com/author/vsubhash>.

FFmpeg installation

FFmpeg now includes three programs — ffmpeg, ffprobe and ffplay. In 2018, ffsserver was removed.

- **Build from source:** It is best if you compile the source code and build the FFmpeg programs on your OS. [FFmpeg.org](https://trac.ffmpeg.org/wiki/) provides guides for several operating systems (<https://trac.ffmpeg.org/wiki/>

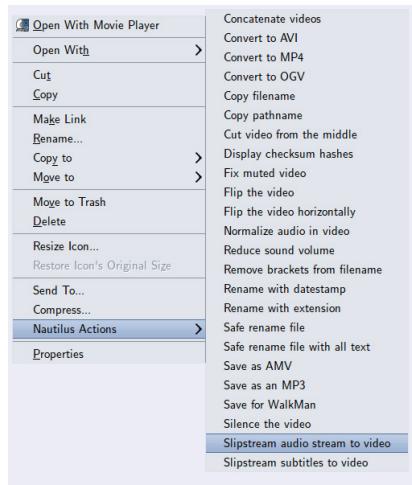


Figure 1: Use shell scripts to automate FFmpeg tasks

[CompilationGuide/Ubuntu](https://trac.ffmpeg.org/wiki/)).

- **Download binaries:** Compiled binaries (executable files of the three programs) are available from <https://ffmpeg.zeranoe.com/builds/>.
- **Statically built binaries:** I am still on Ubuntu 10.10, and I am unable to compile the FFmpeg source code on it. However, I was able to write the book using the latest build thanks to statically built binaries from the site <https://johnvansickle.com/ffmpeg/>. [The grubs in my Ubuntu 10.10 and Ubuntu 20 are unable to boot each other. Anyone who knows the fix can email info@vsubhash.com.]

Concatenate videos

When I wrote my original article, I was an FFmpeg newbie. Cutting and pasting together videos is not as troublefree as the article may have suggested. All input videos have to have the same codec, dimensions and frame rate. Although FFmpeg may try its best, the conversion can fail or the final video

may not be playable.

```
echo "file 'tank-cut-recode.mp4'" >
file-list.txt
echo "file 'tank.mp4'" >> file-list.txt
ffmpeg -f concat -i file-list.txt -c
copy tank-mix.mp4
```

When you try to individually convert the input videos down to a common codec, dimensions and frame rate, and then convert them all into a single file, the lossy algorithm does damage twice. To limit the loss, it is best if you use the raw uncompressed video of each input video. This is rarely the case.

Convert to MP4

While I have no problems with using Ogg video on my Linux systems, my electronic devices do not support it. (This is ironic because they run on Linux.) As I am unable to stream videos because of poor Internet connectivity, I have to convert all non-MP4 videos. The conversion I had suggested in the original article was based on constant bit rate. A better goal is constant quality. FFmpeg's native x264 codec has several presets, ranging from 'veryslow' to 'ultrafast'. You also need to use the constant rate factor option 'crf', whose value ranges from 0 (lossless) to 63 (worst).

```
ffmpeg -i elastigirl.mp4 \
-c:v libx264 -crf 31 -preset
slow \
-c:a copy \
incredible.mp4
```

x264 Presets	x264 Tune	x264 Profiles
ultrafast	film	baseline
superfast	animation	main
veryslow	grain	high
faster	stillimage	
fast	psnr	
medium	ssim	
slow	fastdecode	
slower	zerolatency	
veryslow		
placebo		

Figure 2: Presets available for the x264 codec

Although FFmpeg continues to support several other x264 codecs, the native encoder (libx264) is now the preferred option. An advantage with constant quality is that multiple passes are not required (used to achieve the correct bit rate).

Cut a video

When a raw uncompressed video is converted using a video codec, the compression algorithm creates keyframes to eliminate redundant data. When you cut a video at a particular timestamp, the frames near that location may not have all the information because they are not keyframes. (In other words, clean cuts are not possible with highly compressed videos.) Some frames will have to be eliminated and 'video jumps' become noticeable. The audio and video may also go out of sync.

To limit these problems, whenever you cut a video, fix the timestamps of the frames before further processing.

```
ffmpeg -y -i barbara.mp4 \
-filter_complex \
"[0:v:0]trim=start=0:end=16,
setpts=PTS-STARTPTS[1v];
[0:v:0]trim=start=36:end=44,
setpts=PTS-STARTPTS[rv];
[0:a:0]atrim=start=0:end=16,
asetpts=N/SAMPLE_RATE/TB[la];
[0:a:0]atrim=start=36:end=44,
asetpts=N/SAMPLE_RATE/TB[ra];
[1v][rv]concat=n=2:v=1:a=0[v];
[la][ra]concat=n=2:v=0:a=1[a]" \
-map '[v]' -map '[a]' barb-cut.mp4
```

In this example, the audio and video from seconds 16 to 44 are removed, and the timestamps of the remaining segments are counted again before being provided to the concat filters.

Rotate a video

In the original article, you learned that a video can be rotated in spans of 90°. Values 0 and 3 flip the video in addition to the rotation.

```
ffmpeg -i barb.mp4 \
-filter:video "transpose=1" \
barb-rotated.mp4
```

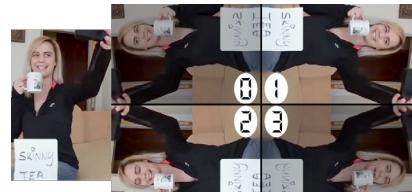


Figure 3: Rotate filter options

Figure 3 shows the transpose filter; filter values 1 and 2 rotate the video right and left. The values 0 and 3 cause rotation but also flip the video.

More discrete levels of rotation are possible with the rotate filter.

```
ffmpeg -y -i barbara.mp4 \
-vf rotate=angle=-
20*PI/180:fillcolor=brown
barb20.mp4
```



Figure 4: This video has been tilted to the left by 20°

If you want to simply flip the video, use *vflip* and *hflip* filters.

```
ffmpeg -i lucas.mp4 \
-filter:v "vflip" \
lucas-upside-down.mp4
ffmpeg -i lucas.mp4 \
-filter:v "hflip" \
lucas-half-crazy.mp4
ffmpeg -i lucas.mp4 \
-filter:v "hflip,vflip" \
lucas-totally-flipped.mp4
```



Figure 5: The flip filters are used to horizontally and vertically flip a video

Normalise audio

You can increase or decrease the volume level in an audio or video file using the volume filter. You set the filter value to a multiple of the input volume or specify the maximum loudness level in decibels.

```
ffmpeg -i low.mp3 -af 'volume=3' high.mp3
```

Adjusting the volume in this manner may not work all the time. Sometimes, the audio is so low that even doubling it does not make a difference. The correct solution involves two steps. First, use the volumedetect filter to determine the loudness of audio samples in the file. Then, apply the volume filter based on what you learnt in the first step.

```
ffmpeg -i low.mp3 -af "volumedetect" -f null -
```

```
[Parsed_volumedetect_0 @ 0x226c100] mean_volume: -32.4 dB
[Parsed_volumedetect_0 @ 0x226c100] max_volume: -17.3 dB
[Parsed_volumedetect_0 @ 0x226c100] histogram_17db: 6
[Parsed_volumedetect_0 @ 0x226c100] histogram_18db: 15
[Parsed_volumedetect_0 @ 0x226c100] histogram_19db: 56
[Parsed_volumedetect_0 @ 0x226c100] histogram_20db: 452
[Parsed_volumedetect_0 @ 0x226c100] histogram_21db: 1676
```

Figure 6: Use volumedetect filter before using volume filter

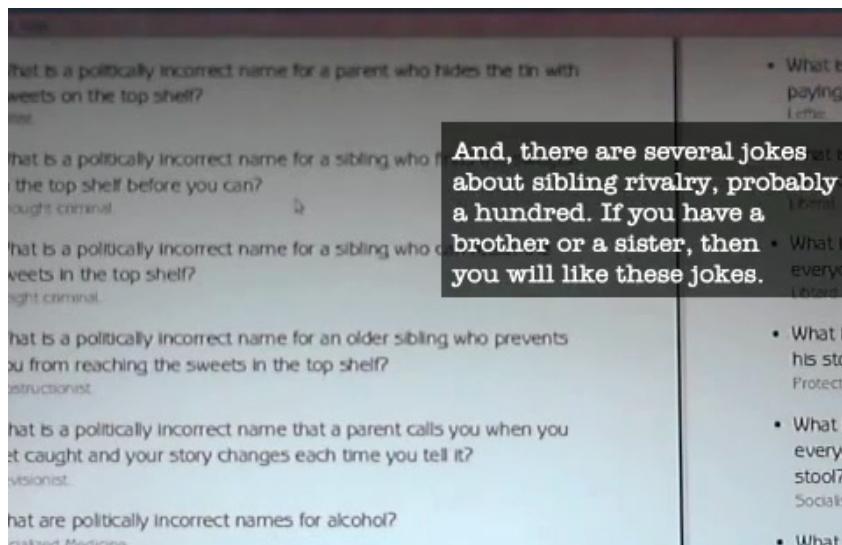


Figure 7: A video with burned-in subtitles

In Figure 6, the volumedetect filter shows that maximum loudness of all samples is at 17dB. The histogram shows the number of samples that would get clipped if the volume was taken any higher than 17. For example, increasing the volume to 18dB would clip the waveform in six samples.

After studying this, it is clear that the loudness needs to be raised to a maximum of 17dB.

```
ffmpeg -i low.mp3 \
       -af 'volume=17dB' -f ogg \
       normalized.ogg
```

Normalisation preserves the original waveform while increasing only the loudness.

Slipstream audio to a video

Sometimes, I encounter a video where the audio has some problems. I usually extract the audio stream to an MP3 file

so that I can examine it in Audacity. After I fix the problem, I export the corrected audio to another MP3 file. Then, I remove the original audio from the video and add this corrected MP3 stream as its audio. I use *maps* to accomplish this.

```
#Extract the audio
ffmpeg -i original-video.mp4 \
       -map 0:1 \
       problem-audio.mp3
```

```
# Fix the problem in problem-audio.mp3
using Audacity and create
# corrected-audio.mp3
```

```
#Replace existing audio with fixed audio
from mp3
ffmpeg -i original-video.mp4 -i
corrected-audio.mp3 \
       -map 0:0 -map 1:0 \
       -codec copy \
       video-with-corrected-audio.mp4
```

In the first command, *-map 0:1* refers to the first input file's or the video's second stream, which happens to be the audio stream. (The *-map* option may be redundant but it helps in debugging more complex commands.) In the second *ffmpeg* command, *-map 0:0* refers to the first input file's or the video file's first stream, which is the video stream. The map in *-map 1:0* refers to the second input file's or the audio file's first stream, which is the corrected audio stream.

Slipstream subtitles to video

Subtitles were very useful to me when I had to create a book-read video for my first book. The audio had issues and I decided to create subtitles and burn them into the video. (I used the GNOME subtitles program to transcribe the video and save to a subtitle file. It has simple keyboard shortcuts for controlling the playback [play/pause/rewind/fast forward] of the video while typing the subtitles. No need to use the mouse.)

I did not like the tiny font used by YouTube so I burned my subtitles into the video. Such subtitles become part of the video and cannot be turned off.

```
ffmpeg -y -i book-read.mp4 \
-filter_complex
"subtitles=transcript.ass" \
-c:a copy \
read-subtitled.mp4
```

While burning subtitles into the video is a failsafe alternative, your first option should be to add a subtitle stream (analogous to a video stream or an audio stream) in the multimedia file container. (MP4, MOV, MKV and other formats are just file containers. The actual codecs can be anything in the wild.) For Matroska videos, you can even add a custom subtitle display font file as a stream.

```
ffmpeg -i sarah.mp4 -i subtitle.ass \
-map 0:v -map 0:a -map 1:s \
-c:v copy -c:a copy -c:s ass \
-metadata:s:s:0 language=eng \
-attach Florentia.ttf \
-metadata:s:3
mimetype=application/x-truetype-font \
sarah-subtitled.mkv
```

In this command, the `-map 1:s` ensures that the subtitles will be from the second file (the SubStation Alpha file). Even if the MP4 file had a

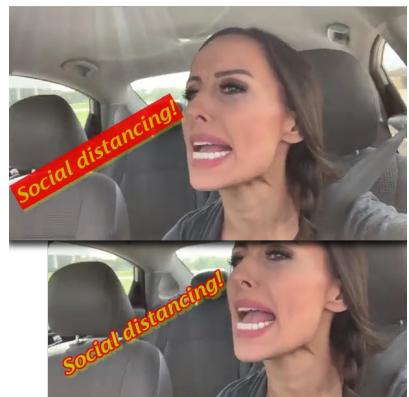


Figure 8: The SubStation Alpha (SSA) subtitle format offers the most features

subtitle stream, it will not be included in the MKV file.

SRT is the most popular subtitle format. SubStation Alpha (SSA) is the most feature-rich subtitle format because it supports a custom font, text colour, outline colour, shadow colour, background box and even rotation. All of this SSA awesomeness is supported only by the Matroska format.

The specification of the endearing but difficult SSA format is available on the <https://www.matroska.org> website (*Technical Info » Subtitles » SSA*). However, I will risk a description here for the style statement.

Style: Name, Fontname, Fontsize, PrimaryColour, SecondaryColour, OutlineColour, BackColour, Bold, Italic, Underline, StrikeOut, ScaleX, ScaleY, Spacing, Angle, BorderStyle, Outline, Shadow, Alignment, MarginL, MarginR, MarginV, Encoding

Name refers to a subtitle display style. You can define and use many different styles, not just the default. The colours are in hexadecimal AABBGGR format! *PrimaryColour*

are automatically used when timestamps collide. *Bold*, *Italic*, et al, are -1 for true and 0 for false. (Yeah, I know.) *ScaleX* and *ScaleY* specify magnification (1-100). *Spacing* is additional pixel space between letters. *Angle* is about rotation (0-360) and controlled by *Alignment*. *BorderStyle* uses 1 (outline and drop-shadow) and 3 (outline box). If *BorderStyle* is 1, then *Outline* represents pixel space width (0-4) of its outline. In the same case, *Shadow* represents pixel space (0-4) below the text and shadow. *Alignment* takes 1 (left), 2 (centre) and 3 (right). If you add 4 to them, the subtitle appears at the top of the screen. If you add 8, it goes to the middle. Then, we have *Margin* from the left, right and bottom edges of the screen. *Encoding* is 0 for ANSI Latin and 1 for Unicode (I think).

I have a Caja Actions Configuration script that changes the style statement in a SSA file to:

Style: Default, Florentia, 30, &H
200CCCC,&H000000FF,&H2200 00EE,
&HAA00CCCC, -1, -1, 0, 0, 100, 100, 0,
30.00, 3, 2, 3, 1, 20, 20, 40, 1. END Penguin



Figure 9: Default SSA styles used by FFmpeg, GNOME Subtitles and the author

is the colour of the subtitle text. *OutlineColour* is for the outline of the text. *BackColour* is the colour of the shadow behind the text. *SecondaryColour* and *OutlineColour*

By: V. Subhash

The author is a writer, programmer, illustrator, cartoonist and humourist (www.vsubhash.com).



“
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:
<http://subscribe.efy.in>

Scan This Code



OR



India Technology Week @ Home

THANK YOU EARLY INVESTORS

We thank the VISIONARY team members of these brands. They chose to believe and invest in our concept of an Online Only Expo-cum-Conference.

They helped us create history. We'll henceforth call them Our Early Investors, and will always remain grateful to them.

AVNET
Reach Further™



Visit 'Online Only' expo-cum-conference from the comfort of your HOME...

 **IndiaTechnologyWeek**
@Home 2020 | SEPTEMBER EDITION
16-17 SEP, 2020



VISIT NOW: <https://Indiatechnologyweek.com>