ClamAV: A Free And Open Source Antivirus Tool

DevOps: An Introduction To Ansible

₹120
ISSN-2456-4885

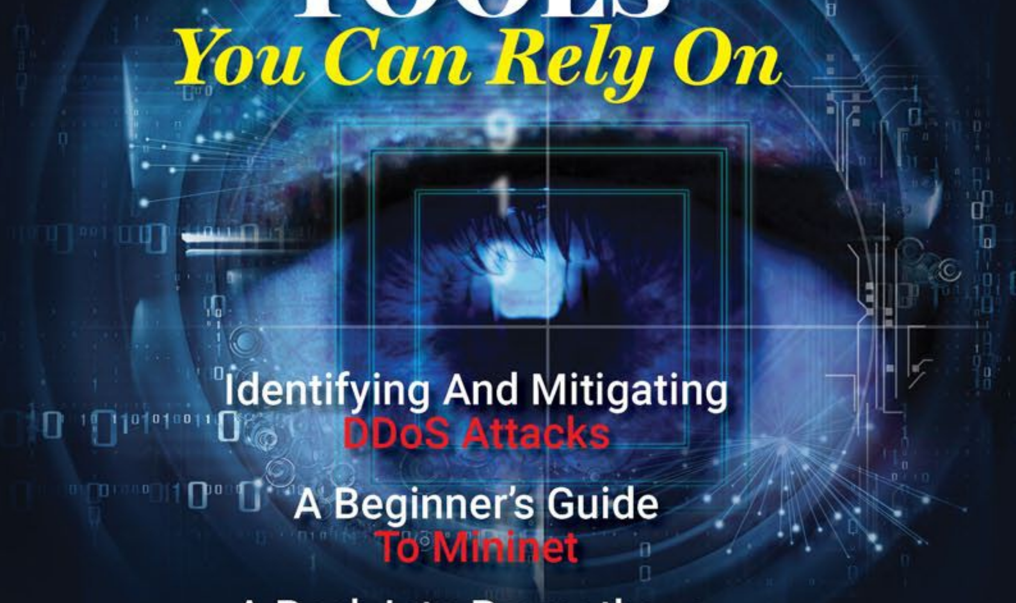
OpenSource ForYou

THE COMPLETE MAGAZINE ON OPEN SOURCE

Volume: 05 | Issue: 06 | Pages: 108 | March 2017

An EFY GROUP Publication

NETWORK MONITORING TOOLS *You Can Rely On*



Identifying And Mitigating DDoS Attacks

A Beginner's Guide To Mininet

A Peek Into Prometheus, A Popular Monitoring Tool

Interview: **Jagdish Harsh,** Founder And CMD, Mobiloitte

Interview: **Mark Collier,** Co-founder And COO, OpenStack Foundation

"It has helped me reach this position."
-Mukesh Ajwani
MD, Electronic Components Trading Firm
"We have grown together in the last 46 years."
-Namit Verma
CMD, UPS & Inverter Manufacturing Company
"I have been in love with it since school."
-Manish Kapur
Sr. Director, Consumer Electronics Giant

electronics
electronics FOR YOU
Printing Organic Parts

CONSOLE
USB
WAN
LAN
OPT1
OPT2
RESET
STATUS
12 VDC

Contents

ISSN-2456-4885

Admin

Developers

FOR U & ME

Open Gurus

REGULAR FEATURES

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: efymum@efy.in

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

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

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

CHINA
Power Pioneer Group Inc.
Ph: (86 755) 83729797, (86) 13923802595
E-mail: powerpioneer@efy.in

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

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

TAIWAN
J.K. Media, Ph: 886-2-87726780 ext. 10
E-mail: jkmedia@efy.in

UNITED STATES
E & Tech Media
Ph: +1 860 536 6677
E-mail: veroniquelamarque@gmail.com

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.

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

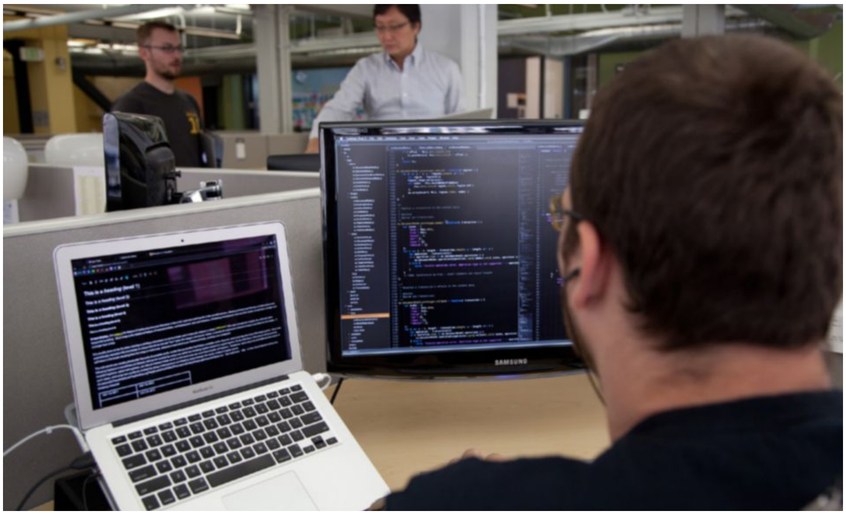
Year	Newstand Price (₹)	You Pay (₹)	Overseas
Five	7200	4320	—
Three	4320	3030	—
One	1440	1150	US$ 120

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.

Compiled by:
Jagmeet Singh

Microsoft builds open source GVFS to virtualise your Git experience

As Git has become the need of the hour in this fast-moving world of code, Microsoft has brought out its Git Virtual File System (GVFS) to deliver an upgraded coding experience. This new solution is available as an open source offering to attract developers.



Microsoft's GVFS is aimed at helping Git clients scale to repositories of any size. This will overcome the challenges of storing a large number of files and is likely to help users preserve their precious codebase.

GVFS virtualises the file system to make it appear as though all the files in your repository are present. But it only downloads files locally the first time they are opened. This helps users save hours of efforts in downloading a single file. Also, users do not need to apply any changes to their IDEs and build tools to make the new solution operable.

Microsoft claims that with its open source GVFS, a clone can be built in a few minutes instead of over 12 hours — a checkout now takes 30 seconds against the two to three hours it does without GVFS, and a status takes four to five seconds instead of 10 minutes.

Though Microsoft's team is internally testing GVFS ahead of its formal release, its open sourced version is already available with all the relevant code and materials on GitHub. This developer-focused version is likely to have some limitations in the initial stages.

"Feel free to give it a try, but please be aware that it relies on a pre-release file system driver. The driver binaries are also available for preview as a NuGet package, and your best bet is to play with GVFS in a VM and not in any production environment," explained Saeed Noursalehi, principal program manager at Microsoft, in a blog post.

The developer team is set to improve GVFS over time. Meanwhile, you can get the upgraded Git with changes for GVFS-backed repositories from the official Git repository. You can also access the protocol extension that the solution uses to integrate third party services.

Mozilla shifts focus from connected devices; lays off entire Firefox OS team

While its rivals like Apple, Google and Microsoft are rigorously working to gain momentum in the nascent IoT space, Mozilla is moving away from the connected devices market. The open source behemoth is laying off the 50-member team behind Firefox OS to reduce managerial expenses.

Mozilla has released a statement to confirm the latest development. "We have shifted our internal approach to the Internet of Things opportunity to step back from a focus on launching and scaling commercial products to one focused on research and advanced development, dissolving our connected devices initiative and incorporating our IoT explorations into an increased focus on emerging technologies," the company stated.

Firefox OS was a dream project from Mozilla to take on Apple's iOS and Google's Android. But a few years after its first commercial release in 2013, the platform left the smartphone world (in December 2015). In September 2016, the company formally stopped development of new Firefox OS smartphones.

Mozilla had subsequently planned to leverage the Firefox OS to impact the IoT space. But now, it is set to step out of that space too.

Technology website CNet reports that Ari Jaaksi, senior vice president and leader of the connected devices division, is among the departing members. Prior to Mozilla, Jaaksi was general manager and senior vice president of the mobile business unit at Intel.

This latest move by Mozilla would indirectly lead to Google boosting its developments around open source IoT offerings. The company already has Android Things as its platform for connected devices, which will persuade developers and hardware makers to build advanced solutions in the coming future.

CloudYuga debuts with a MOOC on containers

As containers are becoming the next big thing in the world of application development, Indian open source enthusiast, Neependra Khare, has launched his training and consulting startup, CloudYuga, to make things easier for Indian students and professionals.

CloudYuga offers training and consultation on various container solutions such as Docker, Rkt and Kubernetes. The first massive open online course (MOOC) that comes from CloudYuga is called 'Container Fundamentals', and it aims to give participants a foundation in container technologies by covering their history as well as building blocks and runtimes.

"The idea behind Container Fundamentals is to build a MOOC platform for container technology, where one can learn everything about containers," said Khare, who was a principal software engineer at Red Hat before establishing CloudYuga. The course delivers a hands-on experience to participants, since CloudYuga's partnership with DigitalOcean enables it to offer on-demand labs for each participant.

Instead of choosing an existing MOOC provider like Udemy, CloudYuga has built its own course platform. "We wanted to have complete control over our training material and student records. The course material will receive regular updates, and we will be interacting with students directly to get feedback and to share information," Khare said.

Those interested do not need to have any prior knowledge about containers to join the MOOC. Basic Linux experience is fair enough. Khare revealed that even with just one MOOC being offered initially, CloudYuga has received a good amount of registrations and community members are supporting the platform.

"Kudos to CloudYuga School for bringing a self-paced learning platform for container enthusiasts to keep abreast of the latest changes in the container space," wrote Docker captain Ajeet S. Raina in a testimonial on the CloudYuga website.

CloudYuga charges US$ 99 for the complete course, which includes video sessions and text tutorials. *Open Source For You* readers can use the code 'CYOSFU40' and get a 40 per cent discount on each sign-up.

TensorFlow brings self-driving to Mario Kart

While many automotive companies are striving to impress customers with their self-driving concepts, the open source software library for machine intelligence, TensorFlow, is bringing the advanced driving model even to video games like Mario Kart.

Open source developer, Kevin Hughes, has deployed Google's TensorFlow to build a self-driving Mario Kart model, which he has called TensorKart. The developer has used the open source N64 emulator mupen64plus to play Mario Kart on his desktop computer, and has taken development support from TensorFlow and cuDNN. Also, he has written a Python program using open source libraries such as wxPython, Pygame and Matplotlib to continuously capture screenshots from the emulator, synchronised with a joystick input. This is the basis of the training set.

"The idea of exploring AI techniques in video games is not new, but what motivated me to do this project was to showcase the complete pipeline of a

Google Earth Enterprise to become open source in March

Google has announced that it is open sourcing its Earth Enterprise software in March 2017. With this development, organisations will be able to deploy Google Maps and the 3D view of Google Earth on their on-premise data centre infrastructure.

Originally introduced in 2006, Google Earth Enterprise (GEE) was being sold by the search giant till 2015. It was designed to let enterprise customers build and host their private, on-premise versions of Google Earth and Maps. But now, Google has given a two-year maintenance period to the customers to help them transition.

After the transition, Google's team is set to release the source code of the Earth Enterprise suite on GitHub under an Apache 2 licence. The open source package will include GEE Fusion, GEE Server and GEE Portable Server. "Open sourcing GEE allows our customer community to continue to improve and evolve the project in perpetuity," Avnish Bhatnagar, senior technical solutions engineer, Google Cloud, wrote in a blog post.

Google Cloud is used to host Earth images. The company will release instructions on using the Google Cloud Storage service with Google Earth before the source code becomes freely available in March 2017. Though a major part of the GEE suite will be open sourced, Google does not have plans to release the code for Google Earth Enterprise Client, Google Maps JavaScript API v3 and Google Earth API.

The imagery and terrain quadtree implementations used in the enterprise products will help third-party developers build viewers that can consume GEE Server Databases. This will enable Google to expand its presence among developers and enterprises.

Open sourcing server software will affect software providers in the same space, such as Esri or ArcGIS Server. But developers would ultimately get a chance to switch to Google's solutions for providing their custom maps and 3D globes.

Linux distro, Tails, leaves 32-bit architecture

Tails, the Linux-based distribution popular for its privacy features, is leaving the world of 32-bit architecture. Tails 3.0 will be the first version to support only 64-bit x86-64 compatible processors.

The developer team has announced that it decided to drop 32-bit support after analysing the statistical data of its users. “It is no surprise that over the last few years, the number of people who use Tails on a 32-bit computer has dropped,” the team said in a statement, adding that at the beginning of 2016, only 4 per cent of Tail users were still using a 32-bit system.

There are many compatibility issues on Tails’ Linux 32-bit version, which its developers have apparently spent a massive amount of time trying to fix. However, the team now wants to focus on more important priorities for 64-bit users.

The Tails team has highlighted two main reasons for opting for 64-bit architecture. First, the new hardware makes it harder for attackers to exploit security vulnerabilities. And, second, the Tails release using a 64-bit Linux kernel is supposed to be more sustainable in the long run.

Tails 3.0 with 64-bit support is scheduled to debut in June 2017. You can check if your computer is compatible with Tails 3.0 by typing ‘uname-m’ in the terminal window. The system generates ‘x86_64’ on the screen to confirm compatibility.

The live version of the Tails Linux distro can be run from an external read-only drive. The Debian-powered platform had gained massive popularity after it was found that whistleblower Edward Snowden used it to protect his identity from investigators in the US government.

machine learning system. I wanted to pick a popular game because I thought it might interest more people and expose them to how machine learning works,” revealed Hughes. The combination of TensorFlow, cuDNN and some open source libraries has enabled Hughes to develop TensorKart’s gameplay.

Hughes did face initial obstacles when he set out on his journey of providing a self-driving model in a video game. “I had to dust off my C programming skills and spend some time remembering all the nuances of C build systems and Makefiles,” Hughes told *Open Source For You*, adding that he also faced initial issues in debugging the originally trained model when it was not moving the car correctly. However, Hughes took support from Stack Overflow and persevered after taking a day off in the middle of the project.

“When I first hit the issue of how to send input back to the emulator, I did not think it would be difficult. I needed to take a day off and appreciate this part of the problem and then approach it with the energy required to fix it properly,” stated Hughes.

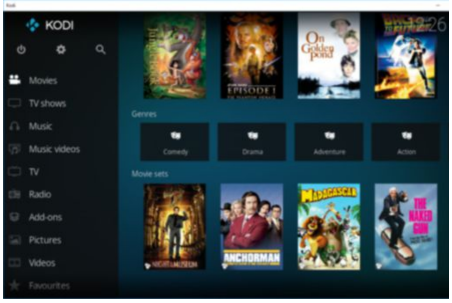
Hughes believes open source is the key to run innovations like TensorKart around the machine learning area. “I have been a believer of open source pretty much since. It became apparent early on how much more efficient open source was; it just felt right. I have benefited a lot from what others have shared and feel good to share my own work,” he said.

Going forward, Hughes plans to expand his TensorKart with new developments. He also expects it to attract more people from the community.

Meanwhile, you can visit GitHub to download the same input plugin that has been enabling self-driving on Mario Kart and build your own advanced gaming experience.

Kodi 17.0 released with 10-foot interface

Team Kodi has announced the release of Kodi 17.0, a.k.a. Krypton. The updated media centre comes preloaded with a 10-foot interface and an enhanced user experience. Kodi 17.0 includes a new default skin called Estuary that has the traditional 10-foot interface. The interface enables the open source media centre to work specifically for televisions, with bold text and special controls for a simple handheld remote control.



For all touchscreen devices, the new Kodi release includes touch-enabled Estouchy. There is also Chorus 2 as the default Web interface.

Apart from the revised interface features, Kodi 17.0 has an upgraded video engine that has undergone a multi-phase rewrite. There are also improvements to stability, audio/video synchronisation, refresh rate switching and decoding/encoding of video. Additionally, the media centre comes with new input stream add-ons that extend support for streaming protocols such as RTMP, MPEG-DASH and SmoothStream.

Team Kodi has not just tweaked the video playback experience but also improved the audio output with performance improvements on tag scraping, support for mood and artist role tags, and increased browsing speed for larger libraries. The live TV and PVR functionality on the Kodi build has also received various improvements. There are now over 15 PVR add-ons and support for the Digital Devices Octopus Net, and asynchronous connections in the back-end.

The Android version of the new Kodi release is standards-compliant with the platform's official audio API. It works with devices that are running on at least Android 5.0 (Lollipop). The release also has DTS-HD, DTS-X, Dolby TrueHD and Dolby Atmos pass through support for devices with AudioTrack v23 or newer.

The Kodi software has built-in support for 4K video and output as well as HEVC, VC-1/WMV 9 and VP9 playback on compatible hardware. You can make all the changes within the latest version of Kodi from its official changelog page. Also, the merged pull requests on the open source project are available from the official GitHub repository.

Interestingly, despite being a highly acclaimed project by the open source community, the latest Kodi is the first release to reach the Windows Store for downloads The software is wrapped in a UWP (Universal Windows Platform) to enable easy installation on Windows 10. It also supports Windows 7 and 8 hardware.

Canonical patches Ubuntu vulnerabilities through new updates

Canonical has released some new kernel updates to fix vulnerabilities within its Ubuntu platform. The versions that are affected by the security issues include Ubuntu 12.04 LTS, 14.04 LTS, 16.04 LTS and 16.10.

Through six distinct security notices on its website, Canonical revealed the vulnerabilities. The company confirmed that the security holes exist across many Ubuntu flavours such as Kubuntu, Xubuntu and Ubuntu MATE, in addition to the original Ubuntu versions.

Ubuntu 12.04 LTS and 14.04 LTS include the security flaw CVE-2016-9555, which is within Linux kernel's SCTP implementation and leads to the platform improperly handling the validation of incoming data, which could result in a denial of service (DoS) attack. The Ubuntu 12.04 LTS build also includes multiple memory leaks within the XFS file system support.

In Ubuntu 16.04 LTS and Ubuntu 16.10, the Canonical team has found two major security issues. The first vulnerability, documented as CVE-2016-10147, is hidden in the asynchronous multi-buffer cryptographic daemon of the Linux kernel. It allows attackers to crash the system via a DoS attack.

CVE-2016-8399, the second issue, is in the Linux kernel's Internet Control Message Protocol (ICMP) implementation. It gives CAP_NET_ADMIN privileges to local attackers to expose sensitive information.

Ubuntu 16.10 also includes the vulnerabilities CVE-2016-10150, CVE-2016-8632 and CVE-2016-9777. These loopholes can either result in a DoS attack to Ubuntu systems, the system crashing or attackers gaining administrative privileges within the host operating system.You can install the latest Ubuntu updates to patch the reported vulnerabilities. Once installed, make sure to reboot your system.

Mirantis containerises OpenContrail within OpenStack ecosystem

Active OpenStack contributor, Mirantis, has containerised OpenContrail. Containerising and supporting OpenContrail will help the California-based company grow into a 'one-stop support shop' for all popular open source technologies used with OpenStack.

OpenContrail is a widely used SDN platform that simplifies OpenStack cloud management for administrators. The rate at which software-defined networking is evolving, administrators may soon have to develop skillsets specifically for OpenContrail. The platform works as an extensible system for cloud networking and network function virtualisation. This would help Mirantis grow its presence in the OpenStack world.

GIMP 2.8.20 brings out several fixes to Linux, iOS and Windows

The GIMP, the popular open source image viewer and editor, has published a new stable release. The GIMP 2.8.20 build has various improvements and bug fixes for all supported platforms, including Linux, iOS and Windows.

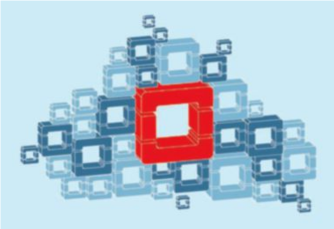
The new GIMP has enhanced the toggling of colour picker mode in the paint tools. On iOS devices, the image editing tool now prompts users to select the preferred language during the first time it's being set up. Its Windows version, on the other hand, has fixed the annoying oscillating switching between input devices. The update also works well on Windows-powered tablets.

In addition to the platform-based changes, the new GIMP version comes with quite a few improvements. There is an improved saving feature for existing *.xcf.gz* and *.xcf.bz* files. Slider handles that were almost not visible while using dark themes have been also fixed. Besides, the updated version has improved stylus tablet support to enhance the image editing experience on tablets.

The GIMP comes pre-installed with various Linux distributions. However, if you are using an iOS or Windows device, you need to manually install it. The update includes an improved installer for Windows and iOS to deliver a faster experience on proprietary platforms.

You can download the GIMP 2.8.20 build from its official download page. The new build comes after six-and-a-half months.

"OpenContrail is an essential project within the OpenStack community, and Mirantis is smart to containerise and commercially support it. The work our team is doing will make it easy to scale and update OpenContrail and perform seamless rolling upgrades alongside the rest of Mirantis OpenStack," said Jakub Pavlik, Mirantis' director of engineering and OpenContrail advisory board member.



Recently, Mirantis acquired TCP Cloud. This acquisition is expected to enable managed services for OpenContrail, Kubernetes and OpenStack. The commercial support by Mirantis will also make OpenContrail compatible with a number of network switches.

According to a recent IDC report, the SDN market will be valued at US$ 12.5 billion by 2020. Mirantis could use OpenContrail to launch itself in this space.

Licensed under Apache 2.0, OpenContrail is built using standard protocols. The open source project has all the necessary components for networking virtualisation. It has an SDN controller, an analytics engine, published northbound APIs and a virtual router.

Google Chrome enables easy access to downloaded files on Android

Google has updated its Chrome Web browser for Android, enabling you to easily access downloaded files directly from the new tab page. The latest version also includes several performance and stability fixes, delivering an improved experience on Android devices.

The Android version of Chrome 56 (56.0.2924.87) has an upgraded tab page that gives you access not just to downloaded files but also to Web pages. The browser also allows you to download article suggestions by long pressing them on the new tab page.

Google has additionally improved the Chrome app to let you quickly use emails, addresses and phone numbers in Web pages by tapping on them. The new update has a list of other improvements too. You can catch all the major changes in the latest version from the Git log. The update will reach the Play store soon.

OpenSUSE Leap 42.2 debuts on major cloud platforms

OpenSUSE is now available across major cloud platforms. With the latest announcement, OpenSUSE Leap 42.2 can be used on Amazon Web Services (AWS) EC2, Google Compute Engine, Microsoft Azure and OpenStack.

The latest OpenSUSE Leap platform has been specifically designed for cloud computing. Users of various cloud platforms can leverage its availability as a virtual image on their cloud instance. Notably, the team behind the open source platform is quite excited about collaboration and support from various cloud providers.

The OpenSUSE Leap 42.2 cloud image is available through the AWS Marketplace. The cloud images for Azure, Google Compute Engine and OpenStack were also released in the past few weeks. "The project has been used extensively for cloud computing, and we are excited that OpenSUSE is now listed in AWS Marketplace. We thank all the cloud providers for working with the OpenSUSE community to make this possible," OpenSUSE chairman Richard Brown wrote in a statement.

The OpenSUSE team has fixed all the loopholes of Tumbleweed while developing Leap 42.2. The distribution is also being talked about as the most advanced release in the OpenSUSE Leap series. The engineers have equipped the platform with cutting-edge tools for cloud users. There are some built-in solutions to ease AWS image uploading and management.

This is not the first time OpenSUSE Leap has arrived on the cloud. A large number of AWS users were previously using the open source platform by installing the OpenSUSE OS images from the last few years. But the official addition to places such as AWS Marketplace has simplified its installation process. Moreover, users on AWS can now also run the Docker container with OpenSUSE cloud images in virtual machines.

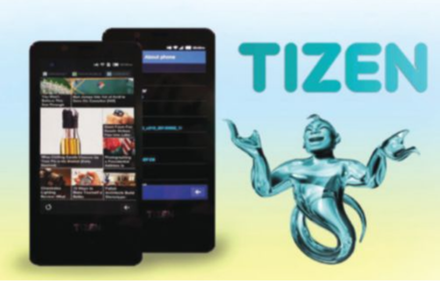
Samsung's Tizen gets .NET support

Months after embracing Microsoft's .NET, Samsung has formally added support for the software framework to its Tizen platform. The new development is for Tizen 4.0, which will be released in September this year.

Samsung had brought out the first preview of .NET Core support and some new Visual Studio tools for Tizen last November. But now, the Korean giant is on the move to make developers familiar with .NET Standard and Xamarin.Forms. This is apparently aimed to expand the app ecosystem of the open source platform. "The first official version of Tizen .NET will be released in September 2017 as a part of Tizen 4.0. Until then, several previews or beta versions will be released in every two or three months so that developers can begin making themselves familiar with Tizen .NET and give early feedback," Samsung's developer team wrote in the official Tizen roadmap page.

The Tizen 4.0 public release with .NET support will be available for both ARM and x86 architectures. Though Microsoft's framework will initially bring new apps to Tizen-based smart TVs, the upcoming support would be expanded across wearables like Samsung Gear smartwatches and even smartphones, over time.

Some initial reports suggest that apart from the range of Samsung's hardware, Tizen will also make its presence felt on development boards such as Raspberry Pi and Artik. This is likely to boost the developer community's interest in Samsung's platform, when building their next apps.



Tizen is already competing against Android and iOS. But the addition of development board compatibility will make the platform a major rival of Android Things, which Google recently launched to support IoT devices. Samsung is already giving away huge cash prizes to developers to attract them to its platform.

Apart from Samsung, Huawei, Intel, Panasonic and Vodafone are members of the Tizen Association. Interestingly, all these players are also supporting Microsoft in its major software developments.

LibreOffice 5.3 comes with online productivity

Expanding its presence in the productivity world, the Document Foundation has released LibreOffice 5.3. This new open source office suite is debuting on the private cloud while being available on Linux, iOS and Windows.

The very first change noticed on LibreOffice 5.3 is its online appearance. First announced last August, the latest LibreOffice version includes the first source release of LibreOffice Online. The new cloud office suite enables basic collaborative editing of documents right in your Web browser.

Fedora 26 to bring out GNOME 3.24 desktop environment

The Fedora project has released a features list of the upcoming Fedora 26. This new version will be shipped with the GNOME 3.24 desktop environment.

The Fedora mailing list is full of the proposals for system-wide changes in the forthcoming release. All new Fedora releases are shipped with the latest GNOME version. Therefore, GNOME 3.24 is debuting on Fedora 26. This desktop environment is scheduled to be published in March 2017.

The new GNOME build comes with numerous features and visual changes. There is an improved control centre with updated user accounts, printer settings panel and online accounts. Also, the desktop environment comes with a provision for a sharing framework and the photo import feature. A new panel to save passwords for key applications is also provided with the latest GNOME.

In addition to the updated GNOME desktop environment, Fedora 26 will offer ownCloud integration within the pre-installed music app. The user interface of the Epiphany Web browser is also due to receive visual changes. Fedora 26 will also have Anaconda LVM RAID implementation to help you create LVM (Logical Volume Management) RAID volumes while installing the OS. Likewise, another proposal suggests the use of KCM (Kerberos credential cache) by default. KCM can be of great use in a containerised environment.

The open source Fedora project is supposed to roll out Fedora 26 on June 6, 2017. It would be the first Fedora version to include OpenSSL 1.1.0 along with new cryptographic algorithms and some major system improvements.

Microsoft adds PowerShell Core to its own repository

Microsoft has just launched PowerShell Core 6.0 alpha on its own Packages repository. All the PowerShell Core releases were previously published on the GitHub repository, but the new change has made its installation much easier. However, the company will continue to support both GitHub and its own repository in the future.

To install the latest PowerShell Core alpha version, you need to register on Microsoft's repository as a super-user. You can update and install the framework using a terminal command after gaining super-user access.

Depending upon the Linux distribution, you need to run the 'sudo apt-get install powershell' or 'sudo yum update powershell' command to update the PowerShell Core package on your system.

It seems that Microsoft is taking extra efforts to improve the experience of PowerShell on Linux. The Redmond company wants to work closely with the community to enhance the experience. Additionally, it is seeking feedback and suggestions from the open source community to improve its approach for PowerShell Core.

Years after its exclusive availability on the Windows platform, Microsoft brought PowerShell to the Linux world last August. The task automation and configuration management framework has been made available as an open source project by the software giant and is compatible with a wide range of Linux distributions such as CentOS, Red Hat Enterprise Linux and Ubuntu.

LibreOffice Online uses the 'core engine' of the open source package. Though it was initially considered as an alternative to Google Docs, the Web package comes as a server service, and needs to be installed and configured manually by adding cloud storage and an SSL certificate. Nevertheless, the online version could help enterprises and large organisations with the public cloud or private cloud technologies. Apart from the cloud-focused release, LibreOffice 5.3 includes the new cross-platform text layout engine. This new offering uses HarfBuzz to deliver a consistent text layout across all platforms. The Document Foundation has additionally provided improvements across languages and alphabets as well as offered a revised *Help* menu with quick links to user guides and community support forums. The updated LibreOffice also comes with improved versions of Writer, Calc and Impress.

The refined Writer has table styling support, enabling users to apply formatting to tables in the document and has a page deck in the sidebar to simplify page setting customisations. It also comes with a new *Go to Page* box to let you jump to another page in the document with just a few keystrokes.

Calc, on the other hand, now has a new set of default cell styles and includes a text entry box to simplify the search for various functions. In a fresh installation, you will notice 'Enable wildcards in formulas' as the default option to improve compatibility with other spreadsheet software like Microsoft Excel.

The new LibreOffice has also added a template selector in Impress to give you a quick-start experience. Besides, there is a new Slide Properties Deck available in the sidebar during slide master mode.

LibreOffice has attracted more than 1,100 new developers. Michaell Meeks of the Document Foundation claimed that, in just the last two years, an average of 300 new people have got active on the source code. "LibreOffice is backed by a fantastic community of developers," Meeks said in a statement. LibreOffice 5.3 is available for free download on its official website. Enterprise deployments can work best on the more mature 5.2.5 version.

Google Chrome on iOS now gets open sourced

After several years of efforts, Google has finally open sourced its Chrome on iOS. The latest development has brought the open source Web browser code to the Chromium project to let developers modify it, just like they can for any of its other versions.

As Apple's iOS is quite conservative when it comes to open source developments, the code for Chrome on iOS was not kept with the rest of the Chromium project until now.

Google also added WebKit support alongside its Blink rendering engine to match the guidelines of iOS. However, all that brought about some 'extra complexities' that would be resolved with the open source version.

"Given Chrome's commitment to open source code, we have spent a lot of time over the past several years making the changes required to upstream the code for Chrome for iOS into Chromium," Google's upstream angler Rohit Rao wrote in a blog post. Google considers that bringing Chrome for iOS to the open source community will increase its development speed. Since all the tests for the browser are available to the Chromium project, they can automatically run on the Chrome for iOS code base with a single execution.

You can access the iOS code from the Chromium repository to compile the newest open source offering. This could help you advance your next Web project.

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

CODE SPORT

Sandya Mannarswamy

In this month's column, we discuss the solutions to some of the questions we discussed in last month's column.

As you know, it is our practice to feature computer science interview questions in the column regularly, for our student readers to prepare for interviews. A few of our student readers have requested me to provide the answers to the questions as well. Given that this column is meant to kindle the reader's curiosity and seek out solutions instead of providing canned answers to questions, I generally do not provide solutions to the questions featured here. So, instead of providing the complete solutions, as requested, it may help the readers to discuss the high level solutions in more detail, while they work out the concrete answers. Hence, in this month's column, we will discuss some of the topics that the featured questions are based on.

One of the questions from last month's column was on the difference between virtual machines and containers. Virtual machines are intended to facilitate server virtualisation, wherein the physical hardware resources are virtualised by a special piece of software known as the virtual machine monitor or hypervisor. This controls the virtual machines that run on top of it. Each virtual machine has its own copy of its operating system. Typically, a VMM or hypervisor runs on top of the host operating system (though it is possible to have a bare metal hypervisor which runs on top of the firmware itself). Each VM runs its own operating system, which is known as the guest operating system. These guest operating systems can be different from each other. On the other hand, containers run on top of a single host operating system, which is why they are sometimes referred to as supporting operating system virtualisation, whereas virtual machines are referred to as supporting server virtualisation. Typically, multiple containers sit on top of a single host operating system, which itself runs on top of a physical server. All the containers share the same host operating system kernel and the libraries.

Now, let us consider the case in which we have a bug in an operating system. If this operating system version was running as a guest OS in a VM environment, the impact of the OS bug would be confined only to that VM running the guest OS. On the other hand, assume that this OS was the host OS on top of which all containers were hosted. All the containers would be impacted by the same bug, since there is no OS isolation in a container environment. In that case, what then is the advantage of containers? The very fact that containers share the same OS components makes them very lightweight and easy to deploy. The container density is much higher than the VM density, given the same physical configuration. A detailed discussion of the security issues of containers can be found in the blog post *https://opensource.com/business/14/7/docker-security-selinux*.

Question (3) from last month's column was about information retrieval. Readers were asked why we need to use inverse document frequency as a measure for relevant documents retrieval, given a query. As we know, TF-IDF is primarily used as the measure for information retrieval. TF stands for term frequency. This is a measure of how frequently a term occurs in a single document. If T1 is a query term and it occurs with a high frequency in document D1, it seems obvious that D1 may be a relevant document when a user is searching for query T1. However, let us consider that our document collection is a set of documents associated with cricket, and our query is 'wisden player'. Now, it is possible that many documents contain the query term 'player' and so its TF will be high for many documents. Hence, this term 'player' may not help us to discriminate the set of documents that are more relevant to the user's query. However, the term 'wisden' is a more significant term as it is likely to be present only in the relevant documents. So we need to get those documents that contain the discriminating term 'wisden' and not include all those documents which contain the non-discriminating term 'player'.

This is where the inverse document frequency comes into play. Inverse document frequency (IDF) is an inverse of the measure of the number of documents that contain the term. Hence, for a discriminating term, since it would be present in fewer documents, its inverse document frequency would be much higher; hence, it can help offset the effect of non-discriminating terms for which the IDF will be much lower. Typically, both TF and IDF are scaled and normalised appropriately. Hence, IDF is typically defined as IDF (term t) = Log (N/Nt), where N is the total number of documents in the corpus and Nt is the total number of documents in which term 't' is present. I leave it to the reader to figure out whether IDF helps to improve precision, recall, or both.

Question (4) in last month's column was about predicting the results of election polls in states. This problem can be analysed in multiple different ways. Let's assume that we want to predict the winning party for a particular state, given that there are multiple parties contesting the election. We also know the past history of elections in that state, as well as various other features collected from past data. This can be modelled as a multi-label classification, where the winner is one of N parties contesting the elections.

On the other hand, if we are considering all the constituencies of the state, can we cluster the constituencies into different clusters such that each cluster represents the set of constituencies won by a specific party. We opt for clustering if we don't have labelled data for the winners in previous elections. In that case, we just want to use the features/attributes of different constituencies and cluster them. However, note that though we get a set of clusters, we don't know what these clusters represent. Hence, a human expert needs to label these clusters. Once these clusters are labelled, an unlabelled constituency can be labelled by assigning the cluster label that is closest to it. So this question can be modelled in multiple ways — either as a classification problem or as a clustering problem, depending on available data for analysis.

Question (8) was about word embeddings, which is the hottest topic currently in natural language processing. Frequency or count based approaches have earlier been widely used in natural language processing. For instance, latent semantic analysis or latent semantic indexing is one of the major approaches in this direction. These approaches have typically been based on global counts, wherein for each word, we consider all the words that have co-occurred with it, to construct its representation. Hence, the entire corpus has to be processed before we get a representation for each word based on co-occurrence counts.

On the other hand, word embeddings are based on a small local context window which is moved over all the contexts in the corpus and, at any point in time, a continuous vector representation is available for each word which gets updated as we examine more contexts. Note that co-occurrence based counts are typically sparse representations since a word will not co-occur with many other words in the vocabulary. On the other hand, embeddings are dense representations since they are of much lower dimensions.

A detailed discussion on the relation between embeddings and global count based representations can be found in the paper 'GloVe: Global Vectors for Word Representation' by Pennington et al available at *http://nlp.stanford.edu/pubs/glove.pdf*. It would be good for the readers to experiment with both Google word2vec embeddings available online (this includes word vectors for a vocabulary of three million words trained on a Google News dataset) as well as the Glove vectors available at *http://nlp.stanford.edu/projects/glove/*. Word embeddings and, subsequently, paragraph and document embeddings are widely used as inputs in various natural language processing tasks. In fact, they are used as input representation for many of the deep learning based text processing systems. There are a number of word2vec implementations available, the popular ones being from the Gensim package (*https://radimrehurek.com/gensim/models/word2vec.html*), which is widely used. It would be useful for the readers to understand how word embeddings are learnt using neural networks without requiring any labelled data. While word embeddings are typically used in deep neural networks, the word2vec model for generating the word embeddings is not a deep architecture, but a shallow neural network architecture. More details can be found in the word2vec paper *https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf*.

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*. Till we meet again next month, here's wishing all our readers a wonderful and productive year ahead!

By: Sandya Mannarswamy

The author is an expert in systems software and is currently working as a research scientist at Xerox India Research Centre. Her interests include compilers, programming languages, file systems and natural language processing. If you are preparing for systems software interviews, you may find it useful to visit Sandya's LinkedIn group 'Computer Science Interview Training India' at *http://www.linkedin.com/groups?home=HYPERLINK "http://www.linkedin.com/groups?home=&gid=2339182"&HYPERLINK "http://www.linkedin.com/groups?home=&gid=2339182"gid=2339182*

Be Up to Date with Technology Strengthen Reading Habits

And, enjoy these FREE Gifts too...

You & Me

YOGG Wrist Band

Your Personal Fitness and Activity Tracker

Worth ₹2999

Worth ₹1599

Sound Pot

Portable, Aesthetically Designed and Rubberised Premium Finish Bluetooth Speaker

Techies

Raspberry Pi 3

Latest version of Raspberry Pi is a low cost, credit-card sized computer

Worth ₹3300

Wireless Headphone
Worth ₹1999

USB Car Charger
Worth ₹499

5200mAh Power Bank
Worth ₹1999

IoT Starter Kit
Self Learning Kit for Hobbyist, Enthusiast and Students

Worth ₹2000

Arduino Uno
Worth ₹890

Smart WiFi Module
Worth ₹725

ORDER FORM

Magazine	Duration	Cover Price (₹)	You Pay (₹)	Gifts For You & Me (Please Tick Any One)	Gift for Techies
Open Source For You with DVD	1 Year	1440	1440	□ 5200mAh Power Bank - worth Rs 1999	□ Arduino Uno worth Rs 890 + Smart WiFi Module –worth Rs 725
	2 Years	2880	2880	□ Wireless Headphone - worth Rs 1999 + USB Car Charger – worth Rs 499	□ IoT Starter Kit – worth Rs 2000
	5 Years	7200	7200	□ Yogg WristBand - worth Rs 2999 + Sound Pot — worth Rs 1599	□ Raspberry Pi 3 – worth Rs 3300

Free e-zine Access With Every Subscription | **To Subscribe Online, visit: http://subscribe.efyindia.com**

Name__________ Designation__________ Organisation__________

Mailing Address__________

City__________ Pin Code__________ State__________ Phone/Mobile__________ Email__________

Subscription No. (for existing subscribers only__________. **I would like to subscribe Open Source For You 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)*

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: **011-26810601-03** | Fax: **011-26817563** | e-mail: **info@efy.in** | **www.efy.in**

EFYGROUP
Technology Drives Us

Terms & conditions: # This offer is applicable for new subscriptions and renewal by existing subscribers. # The rates are valid for subscribers within India only. # Please allow 4-6 weeks for processing your subscription and 6-8 weeks to send your gifts # Replacement copies will be sent only if intimation of damaged / non-receipt of copies is received within 15 days of publication # If you are not satisfied with the magazine or our services, we shall refund the balance amount after three months. # Disputes, if any, are subject to exclusive jurisdiction of competent courts and forums in Delhi/New Delhi only.

NEW PRODUCTS

Price:
₹ 999

High capacity power bank from UIMI

Electronic devices manufacturer, UIMI, has launched the UIMI 8, a power bank with 15,600mAh battery capacity. The device has a fine stainless steel metallic finish and an ultra-sleek compact design. It sports high quality li-ion cells, which enable users to charge their devices four to five times, without charging the power bank.

The UIMI 8 comes with innovative FitCharge technology, which allows users to simply connect their device to the power bank for charging, without the need to switch it on. It offers a single input port for charging the power bank and dual output ports for charging other devices.

It also provides protection against overheating, short-circuits and overcharging by automatically disconnecting the device after full charge. It features an LED torchlight and battery-level indicator to keep a track of the remaining charge on the battery.

The UIMI U8 power bank is available in black and gold, via online stores.

Address: UIMI Technologies, F-16,
Sector-6, Noida, Uttar Pradesh
201301; Ph: 91-120-4552102

Smartwatch with built-in camera from Zebronics

Price:
₹ 2,199

Computer and audio peripherals manufacturer, Zebronics, has launched its smartwatch – Smart Time100. The device comes with a 3.9cm capacitive touchscreen display with a screen resolution of 240 x 240 pixels and a built-in 0.3 megapixel camera.

The smartwatch supports Bluetooth 3.0 and is compatible with nearly all Android devices. It comes with 64MB RAM and 32MB internal memory, which is further expandable up to 32GB via microSD card. It has a built-in 280mAh battery with a standby time of up to 150 hours and offers talk time of up to 3 hours via Bluetooth.

The device also offers a SIM slot, where a microSIM can be inserted to use it as a standalone device. This affordable smartwatch offers additional features like a pedometer, sleep monitor, sedentary reminders, anti-loss alert and a calculator.

The Zebronics SmartTime 100 is packed with a ZEB-BH502 Bluetooth headset for answering calls. It is available online and at retail stores.

Address: Zebronics India, Pillar No
211, Patel Nagar, Delhi – 110008; Ph:
07654107727

Waterproof Bluetooth earphones from Panasonic

Japanese multinational electronics company, Panasonic, has launched Bluetooth sports earphones, called the RP-BTS 50. These ultra-light Bluetooth earphones are rugged and comfortable for outdoor activities such as running, cycling, etc.

The waterproof device offers hassle-free use even while sweating or when in the rain. It offers premium 12mm drivers for full rich bass and natural treble. It comes equipped with embedded blue LED lights on the edges, which act as a safety measure when used at night.

The RP-BTS 50 headphones come with a flat cable and a 12mm removable driver, which is compatible with aptX and Advanced Audio Coding (AAC).

The device offers quick-charge, and up to 70 minutes playback with 15 minutes of charge. It is compatible with all Android and iOS devices.

Price:
₹ 8,999

The Panasonic RP-BTS 50 is packed with ultra-soft small, medium and large ear pads, a USB charging cord, a travel case and is available online and at retail stores.

Address: Panasonic India Pvt
Ltd, 12th Floor, Ambience Tower,
Ambience Island, NH-8, Lane No.
V-40, DLF Phase-3, Sector 24,
Gurugram, Haryana – 122002

Phablet with Google Tango AR technology from Lenovo

Price:
₹ 29,990

Lenovo has introduced what it claims is the world's first Tango enabled AR phablet, the Lenovo Phab 2 Pro, in India. Google's Tango technology offers a set of sensors and software that detects and maps surroundings to enable a host of smartphone augmented reality (AR) experiences.

The Phab 2 Pro comes with a 16.2cm (6.4 inch) touchscreen display with a resolution of 1440 x 2560 pixels, which adapts to variable lighting conditions like sunlight or light reflections. The device is designed with an 8.9mm aluminium unibody and 2.5D curved glass, with a fingerprint sensor on the rear.

The phablet is powered by a 1.8GHz octa-core Qualcomm Snapdragon 652 processor, specially optimised for Tango, and runs on Android 6.0 Marshmallow with 64GB internal memory expandable up to 128GB. Backed with a 4050mAh non-removable battery with 2.4x turbo charging, the phablet offers a 16 megapixel primary camera on the rear and an 8 megapixel front camera.

The Lenovo Phab 2 Pro is a single SIM (GSM) smartphone that offers connectivity options like Wi-Fi, GPS, Bluetooth, NFC, and 3G/4G with sensors including a compass, magnetometer, proximity sensor, accelerometer, ambient light sensor, etc.

The device is available in champagne gold and gunmetal grey, online and at retail stores.

Address: Lenovo India Pvt Ltd, Vatika Business Park, 1st Floor, Tower A, Sohna Road, Sector 49, Gurugram – 122018; Ph: 124-3055600

A pocket-friendly tablet from Ambrane

Electronic product manufacturer, Ambrane, has recently unveiled its affordable tablet, the AQ11. The value driven tablet promises efficient performance in terms of features and applications.

The tablet features a 25.4cm (10 inch) IPS display in a sleek and elegant design, which is easily portable and comfortable for use. The device is armed with a quad-core processor and 1GB RAM allowing multitasking, switching between various applications, etc. It runs on an optimised version of Android 5.1 Lollipop with 8GB internal memory, further expandable up to 32GB via microSD card. It is equipped with a 5000mAh Li-ion battery offering talk time of up to 26 hours and standby time of 820 hours.

The device sports a 5 megapixel rear camera with flash support and a 2 megapixel front camera for selfies and video calling.

The Ambrane AQ11 offers connectivity options like 3G, Bluetooth, Wi-Fi, GPS, etc. The tablet is available online and via retail stores.

Address: Ambrane India, C-91/7, Wazirpur Industrial Area, Delhi – 110052; Ph: 011-48089900

Price:
₹ 7,999

The prices, features and specifications are based on information provided to us, or as available on various websites and portals. OSFY cannot vouch for their accuracy.

Compiled by: Aashima Sharma

Anil Seth

Modelling the Shortage of Change Using Netlogo

Last month we explored Netlogo, the multi-agent programming environment which simulates natural and social phenomena. In this article, the author skilfully uses an existing real-world situation, namely, the shortage of change due to demonetisation, to set up a Netlogo model.

We have all recently experienced the shortage of cash or change. Since my expectations of the cash availability situation post demonetisation did not turn out to be correct, I wanted to see if a simple model could be created in Netlogo, to model the shortage of change.

The model

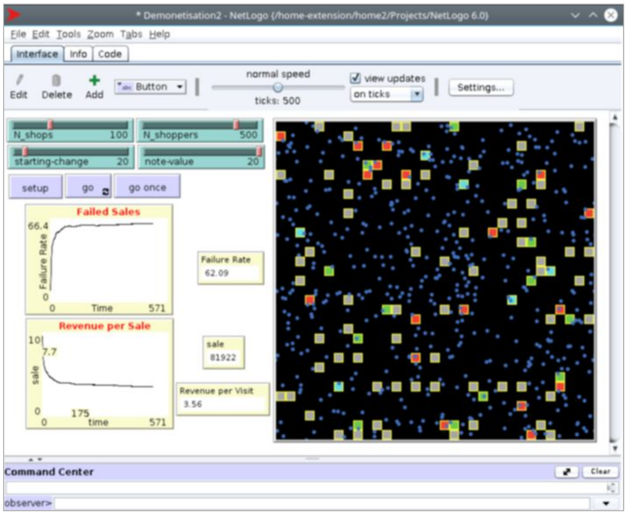
The simple model I chose was as follows:

- Our world consists of shops, shoppers, coins and notes.
- Each note is equal to note-value coins.
- There are a fixed number of shops, N_shops, at fixed locations.
- Each shop starts with starting-change coins.
- There are a fixed number of shoppers, N_shoppers, who move a step on each tick.
- Each person starts with no coins but an infinite number of notes.
- If a person lands at a shop, (s)he makes a purchase of a random value between 0 and note-value excluding the end points.
- Each person pays with coins if (s)he has enough and does not hold onto them.
- Each shop returns the change if it has enough, even if the sale is for a small amount.
- If neither the shop nor the shopper has an adequate number of coins, the sale fails.
- You can vary the values of N_shops, N_shoppers, starting-change and note-value to study the rate of failed sales.

The code

The interface screen is shown in Figure 1. This includes the values you can set for the number of shops and the number of shoppers in this world. You also configure the amount of change each shop has and the number of coins that add up to the value of a note.

You need to define 'turtles' called shops and shoppers, and various variables for the model. The global variables keep track of the total value of sales, the number of sales and the number of sales that failed because there was no change.



* Demonetisation2 - NetLogo (/home-extension/home2/Projects/NetLogo 6.0)

normal speed

ticks: 500

☑ view updates: on ticks

N_shops 100

N_shoppers 500

starting-change 20

note-value 20

Failure Rate 62.09

sale 81922

Revenue per Visit 3.56

Figure 1: Interface for the demonetisation model

```
breed [shops shop]
breed [shoppers shopper]
shops-own [shop-coins]
shoppers-own [coins]
globals [sale no-change count-sales]
```

The set-up code sets up the shops at non-overlapping locations. Each shop is a square. The shoppers are created at random locations and shaped as dots.

```
to setup
  clear-all
  set sale 0
  setup-shops
  setup-shoppers
  reset-ticks
end

to setup-shops
```

```
  ask n-of N_shops patches
  [sprout-shops 1 [
    set shape "square"
    set colour grey
    set pcolor yellow
    set shop-coins starting-change
    set no-change 0
    set count-sales 0
    ]
  ]
end

to setup-shoppers
  create-shoppers N_shoppers [
    setxy random-xcor random-ycor
    set color blue
    set shape "dot"
    set coins 0
  ]
end
```

The modelling code runs for 500 ticks. Each shopper is asked to move forward a step and check if there is any shop in that location. If there is a shop, a sale is attempted. The value of each sale is selected at random by a number that ranges between 1 and the note value. If a shopper has enough coins, (s)he pays with the coins. If (s)he does not have enough coins, (s)he pays with a note and the shop returns the change if it has enough coins. The sale fails if that is not possible.

```
to go
  if ticks >= 500 [ stop ]
  ask shops [ set color grey ]
  ask shoppers [
    forward 1
    if any? shops-here [
      make-sale
    ]
  ]
  tick
end

to make-sale
  let value max (list 1 random note-value)
  ifelse coins >= value
  [ set coins coins - value
    ask shops-here
    [ set shop-coins shop-coins + value
      set color green
    ]
    set count-sales count-sales + 1
    set sale sale + value
  ]
  [ ask shops-here
   [ ifelse shop-coins >= (note-value - value)
     [ set shop-coins shop-coins - (note-value - value)
       ask myself [ set coins coins + (note-value - value)]
       set sale sale + value
       set count-sales count-sales + 1
       set color cyan
     ]
     [ set no-change no-change + 1
       set color red
     ]
   ]
  ]
end
```

During the running of the model, the interface shows the failure rate and the revenue per sale attempted.

Running the model

The objective is to code. But you can verify that some of our intuitive perceptions are reasonable. You may choose the number of coins in a note to be 20 (modelled on ₹ 2000/₹ 100). Let's suppose there are 100 shops and 500 shoppers, and the shops start out with change of a 100 coins each. We expect that the failure rate will be low. It stabilises at about 5 per cent with an average earning per visit of nine coins. Now let's suppose that the starting change of each shop is 20 coins. How badly would the sales be affected? In our model, the failure rate reaches over 60 per cent and the average earnings per visit drop to about 3.5 coins.

You can explore and play around with the model.

The keyword 'ask' is the critical construct which allows you to code for a specific breed of turtles or patches (the location/coordinates in our world).

The 'trickiest' code is the creation of N shops located randomly. In this case, you ask any N patches to 'sprout' a shop (ask n of N patches ….).

This is a very useful feature if you need to have only a random subset of turtles to do something.

The key takeaway is that surprisingly little code is needed to start exploring complex phenomena and understand the statistical implications of even simple interactions. END

By: Dr Anil Seth

The author has earned the right to do what interests him. You can find him online at *http://sethanil.com* and *http://sethanil.blogspot.com*, and reach him via email at *anil@sethanil.com*.

FCOOS Delivers a Rewarding Open Source Experience to its Customers and the Community

FCOOS (Flying Concepts On Open Source) is one of the leading open source based IT infrastructure service providers based in Bengaluru with operations across the world. The company's core expertise is in IT infrastructure management and it has productised services related to firewalls, email servers, the cloud and VoIP gateways.

Flying Concepts On Open Source

As its name suggests, FCOOS' core expertise is in open source technologies and open standards. The company believes that its religion is open source, and operates in service domains that are specialised in IT infrastructure.

FCOOS positions itself between the community and its customers. Led by founder, CTO and community coordinator, Sandeep Athiyarth, the company adds value to both the community and its customers by supporting and practising open source models. It contributes to the community financially by sharing nearly 5-10 per cent of the projects, delivering profits in various ways. This comes from either direct contributions, providing training or through various partnership subscriptions. Also, the Bengaluru-based company maintains forums and submits bug reports to the communities that it is directly involved with. Currently, the company is not involved in any development activities but is in the process of building up expertise to contribute on various fronts.

IT infrastructure service delivery models

FCOOS is mainly a team of systems, servers and network administrators. Shrinivas Rao, director of IT infrastructure operations, leads different IT teams that jointly serve three types of delivery models, namely, onsite, visit based and remote.

FCOOS' resident engineers work on the onsite model, supporting enterprise infrastructure with advanced monitoring tools and remote expert teams. The company also uses modern open source tools to anticipate any issues, take the necessary action when required and adopt proper change management methods.

The visit model is based on support. This is a tailor-made option for small and medium enterprises that do not require an in-house support team to manage their IT infrastructure. FCOOS is also leveraging its remote team, IT infra management team and monitoring tools manned round the clock to power its visit based support.

The third model comprises the end-to-end remote management of IT infrastructure and applications. Here, the remote team by FCOOS is available almost round the clock to govern, administer, manage and monitor. These will be mainly based on open source and some proprietary applications as well. There is a team of engineers certified by companies like Red Hat, Cisco, Checkpoint, Oracle and Microsoft to support various deployments.

Firewall and security services

The firewall service by FCOOS is currently centred on the world's most trusted and popular open source firewall —pfSense. The company has officially partnered with the pfSense team and Negate. Lead by Nilesh Bosamiya, the firewall team operates as an authorised value reseller of Netgate hardware with pfSense, and it sells and supports its homegrown hardware using the same open source firewall.

Additionally, FCOOS has network auditing and vulnerability scanning services under its security services team. These also come with the open source tools, OpenVAS or Nessus.

Services targeted at the cloud and devOps

FCOOS has a dedicated cloud team supporting customers from across India and abroad. Led by Balaji Murugan, the sales team for cloud services has officially partnered with Amazon Web Services, and receives support directly from an AWS-dedicated sales and technical team.

The AWS-certified team provides support on top of AWS infrastructure with various Web applications like Apache, Tomcat and Node JS. FCOOS also has an expert team focused on MySQL, PostgreSQL and MongoDB administration. It has a considerable customer base with Magneto deployed for various e-commerce sites. The company is an official partner of Docker, and has customers with automation engines with Puppet and Chef.

Partnerships with Elastix and Red Hat

FCOOS is an official partner of Elastix and serves customers across India, improving their ROI by using open source for their voice communication requirements. The company provides migration and support services on top of Elastix-based Asterisk solutions. It is also an authorised Elastix reseller and service provider.

In addition to Elastix, FCOOS has recently started concentrating on Red Hat products and services. The company is considering the SaaS (Software-as-a-Service) model to let customers pay only for monthly subscriptions, thus avoiding any capital investment.

ERPNext and Bugzilla

Open source is a religion for FCOOS. Recently, it has partnered with ERPNext, another great open source product team and a vibrant community, and started using ERPNext at its own level and proposes to do the same for its customers.

FCOOS has additionally started the migration of Tally and other proprietary ERP systems to ERPNext, which is progressing at a much faster pace than expected. Since the company's inception, it has partnered with the Bugzilla project and has contributed to it in various ways.

Approach to proprietary software products

FCOOS has a passion for and the expertise in open source. This does not mean that the company opposes proprietary software or closed development models, but respects them as other technology models and considers them to be important options for customers.

When FCOOS takes up any project that involves supporting proprietary software, it quickly begins specialising in those requirements. This ensures value to the customer, who depends only on a single vendor but still enjoys the required support. Still, if there is an option to migrate an application to the relevant open source platform, FCOOS proposes the idea and offers a better ROI to the customer.

Hosting services

Along with delivering open source solutions, FCOOS offers two hosting services. One is the hosted email services with iRedmail, and the second is backup solutions with ownCloud and BackupPC. The company is not into the commodity market of mailboxes. Instead, it provides these services as value addition to its IT infrastructure support customers. Still, the Bengaluru based company is able to match the price of ₹ 75 per mailbox, which is comparatively low cost. The same goes for its backup services as well.

In the SMB market, there is a lot of confusion about backups and how to take them in the existing IT infrastructure. FCOOS evaluates an SMB's requirements and takes full responsibility for its backups by providing compute and storage along with services. END

Be FIT For BUSINESS

Portronics

YOGG Wrist Band

Your Personal Fitness and Activity Tracker

MRP ₹2999

Worth ₹1999

Wireless Headphone

Ultra Modern, Wireless & Foldable Headphone with Discreet Invisible Microphone

Combo Pack, Set of 4
12W LED Bulb

Set of 5 0.5W Night lamp

Worth ₹2700

Velocity Power Bank

Worth ₹2999

UFO Home Charger

Worth ₹1299

Combo Pack, Set of 3 – 10W LED Bulb with set of 2, 0.5 Night Lamp

Worth ₹1500

One 9W LED Bulb + Three 0.5W Night Lamps

Worth ₹700

ORDER FORM

Magazine	Duration	Cover Price (₹)	You Pay (₹)	Gifts For You & Me (Please Tick Any One)	LED Lights
Electronics Bazaar	1 Year	1200	1200	□ UFO Home Charger – worth Rs 1299	□ One 9W LED Bulb + Three 0.5W Night Lamps – worth Rs 700
	2 Years	2400	2400	□ Velocity Power Bank – worth Rs 2999	□ Combo Pack, Set of 3 – 10W LED Bulb with set of 2, 0.5 Night Lamp - worth Rs 1500
	5 Years	6000	6000	□ Yogg WristBand - worth Rs 2999 + Wireless Headphone - worth Rs 1999	□ Combo Pack, Set of 4 – 12W LED Bulb with set of 5, 0.5W Night lamp - worth Rs 2700

Free e-zine Access With Every Subscription | To Subscribe Online, visit: http://subscribe.efyindia.com

Name__________ Designation__________ Organisation__________

Mailing Address__________

City__________ Pin Code__________ State__________ Phone/Mobile__________ Email__________

Subscription No. (for existing subscribers only__________. **I would like to subscribe Electronics Bazaar 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)*

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: **011-26810601-03** | Fax: **011-26817563** | e-mail: **info@efy.in** | **www.efy.in**

EFYGROUP
Technology Drives Us

Terms & conditions: # This offer is applicable for new subscriptions and renewal by existing subscribers. # The rates are valid for subscribers within India only. # Please allow 4-6 weeks for processing your subscription and 6-8 weeks to send your gifts # Replacement copies will be sent only if intimation of damaged / non-receipt of copies is received within 15 days of publication # If you are not satisfied with the magazine or our services, we shall refund the balance amount after three months. # Disputes, if any, are subject to exclusive jurisdiction of competent courts and forums in Delhi/New Delhi only.

“A SINGLE SILVER BULLET CANNOT MEET ALL THE CHALLENGES IN THE IoT SPACE”

The Internet of Things (IoT) has become the next big thing in the technology world. But there are some big challenges to be overcome. ***Jagdish Harsh, founder and chief managing director of Mobiloitte***, an IT company based in New Delhi, Singapore, UK and USA focusing on bots, apps, digital and IoT, discusses these challenges with ***Jagmeet Singh*** of ***OSFY*** and highlights the potential of the growing IoT space for Indian developers. Edited excerpts...

Q How has IoT evolved in India?

IoT has evolved significantly across the globe. You can find some great examples of this evolution in Singapore, Denmark and Spain. And this same pace of growth is catching on in India with the government’s clear focus on digitisation and the huge budget allocation around connected healthcare. A few focused companies are taking the initiative in this space, on their own.

There is a large pool of well-trained medical professionals available in India, which has helped to enhance the potential of IoT developments in the healthcare sector. Also, the government has provided policy support to IoT enablers by reducing excise and custom duties, and offering exemptions in service tax.

Having said that, manufacturers, entrepreneurs, software/hardware specialists and domain experts from all sectors need to collaborate and work hand-in-hand to set the right expectations. Apart from harnessing technology around IoT, empowering people in rural areas by making the solution platform affordable is likely to be the key to drive the growth of connected devices.

The medical tourism market in India is reported to have had a turnover of around US$ 3.9 billion in 2016 and is predicted to hit US$ 8 billion by 2020. This is a major reason for manufacturers to develop new healthcare-focused IoT devices. Some new drug testing laboratories are also being established to advance medical services and help the government develop the country as a global healthcare hub in the near future.

The turnover of the global IoT market is expected to cross US$ 220 billion by 2020.

About Mobiloitte

Mobiloitte's strength lies in its ability to know where IoT applications are needed as well as in finding faster and cheaper ways to connect with devices. The company also focuses on greener manufacturing and developing solutions with completely remote mobile device management (MDM), thus reducing device management complexity.

Instead of just delivering cheaper solutions, Delhi-based Mobiloitte helps enterprises make affordable IoT solutions without compromising on security and interoperability. It develops products for predictive prevention of health issues, and to improve machine efficiency and reliability.

Q How is IoT enhancing the healthcare sector in India currently?

The Indian government is increasing its spending on healthcare, and closing the infrastructure gap as well as addressing the workforce scarcity. But to bring IoT to the healthcare sector, many big organisations have just started contributing with end-to-end patient engagement platforms. Wearables are also leading to innovations that enhance the fitness of a wider proportion of the population.

Apart from some large companies, startups are also actively researching and partnering up with NGOs to improve the health of the masses, using IoT.

Since 2012, the Indian health sector has witnessed active participation by IoT developers, resulting in solutions that address the problems arising from the 21st century's stress driven lifestyle. Wrist bands that add a dash of wearable fun to the fitness process are a huge hit among the health-conscious youth. Sensor-powered pendants that can detect falls are also available to support elderly people.

Q What are the big challenges in developing apps for the IoT world?

Just like building solutions in any other nascent space, creating apps for IoT also involves some major challenges.

Developers need to focus on network latency, determinism and bandwidth when building IoT apps for precision-based machines. If the timing is off, even for a millisecond, the entire production floor could fail. Likewise, IoT systems need to be adaptive and scalable through software. Having black box systems that do not communicate well together are of no use in the connected space.

IoT applications can be built on tens of thousands of sensor nodes, but all this increases threat surface areas by orders of magnitude. Therefore, developers need to implement security at each and every level of their code. Industrial systems also have to be continually modified and maintained to meet changing requirements. Additionally, an effective platform-based approach should focus not on hardware or software but on the innovation within the application itself, and provide enough flexibility to the system to evolve and adapt.

Q How does your team resolve challenges in building IoT apps?

A single silver bullet cannot meet all the challenges in the IoT space. However, we consider that the challenges in developing apps for the connected space are forecastable —their redundancy is built-in and alliances with trusted and highly skilled organisations should be formed. We also prefer the establishment and formalisation of checklists and globally acceptable protocols in order to align different tools, technologies, people and processes.

Q Do you leverage any open source technologies while developing apps for IoT hardware?

The AllSeen Alliance, DSA, Eclipse IoT (Kura), and Open Connectivity Foundation are some of the promising open source technologies and platforms for connected devices. However, no single technology fits all solutions.

Hence, we believe in creating a hybrid architecture, which is the consolidation of the best of the breed from a plethora of open source technologies, as a one-stop solution for most IoT needs.

Q Is it easy to create a network of IoT devices through open source?

This would vary from case to case. While open source solutions can address some challenges, proprietary solutions can address other use cases, and several other use cases can be addressed via a mix of open source and proprietary solutions. As a thumb rule, the best-of-breed solutions need to be leveraged to create a seamless IoT platform that is universally acceptable, works in real-time, is cost effective and predictive.

Q Can a simple app developer be an enabler in the IoT world?

Yes, why not? The transformation of a traditional app developer to a developer for the IoT world requires only a few steps. We make this possible at Mobiloitte and create a pool of developers by training them, monitoring their progress on a weekly basis, and filling in their skill gaps by stepping in and providing thought leadership.

We have our own CoE (Centre of Excellence) managed by the Training Resources Group (TRG) and led by our CTO that continues to conduct research on new and upcoming technologies, creates reusable components, and develops zero build reusable frameworks as well as ready-to-use code snippets and documents. All these are made available, step by step, on the Intranet to train new employees. We also have a team that specialises in the IoT space, apart from many other complex and intelligent applications and programs. Hence we're able to offer an environment of continuous learning to traditional developers.

> Developers need to focus on network latency, determinism and bandwidth when building IoT apps for precision-based machines.

Q Is it possible for a developer to reach out to Mobiloitte for recruitment merely on the basis of some prior app development skills?

At Mobiloitte, we have a policy for social upliftment, which entails hiring B. Tech and M. Tech graduates right from their college campuses located in not-so-developed places like Gorakhpur or Bareilly and from the underdeveloped regions of Odisha, Uttaranchal and Punjab. They are handpicked through a well-designed, on-campus screening process and then brought to New Delhi.

We provide weekly training on trending technology topics to help them on domain- and technology-specific topics. Training is conducted by our Training Resources Group (TRG) members who first assess, evaluate and identify areas for improvement before providing the relevant education.

We also have a well-defined programme for people with physical disabilities to embrace Mobiloitte and become enablers not only in their professional lives but also in their personal lives. We believe that physical disability has nothing to do with intellectual capabilities.

Q What are the prime security areas that you focus on while developing apps for connected devices?

Development around IoT, where multiple devices are connected with multiple protocols and platforms, comes with a new set of security threats. Therefore, we focus on minimising the collection of personal data. We believe that personally identifiable information (PII) should be collected only to the extent that is required. This collected data must be encrypted and privilege access policies set, by default. We also ensure information flow enforcement and secure physical access control on IoT equipment to prevent access by malicious users.

You cannot completely secure the apps for connected devices, but limiting multi-user access and releasing patches based on the behavioural analysis of users helps to reduce the surface threats for attacks. There are many major security threats that we are continuing to evaluate, assess and monitor.

Q Do you think security is one of the vital concerns for an IoT app maker like Mobiloitte?

Yes, security is the primary concern for any IoT-focused organisation including Mobiloitte. Security threats are, in fact, one big reason why adoption of IoT is not gaining the momentum it should.

As more modern medical devices are deployed, adding to the growing collection of IoT connected devices, many healthcare professionals have found that with these advanced devices, come more advanced cyber threats as well.

There is a need to design protection services to reduce attacks in the IoT space, even while the deployed detection services receive data from healthcare applications, devices and networks and analyse it for any anomalies. With the aid of defence mechanisms, these detection services should also help health devices survive all mass security attacks.

Q How do you ensure that your clients' intellectual property (IP) is secured?

We at Mobiloitte have designed and successfully delivered very innovative products including IoT-based solutions, as well as solutions in augmented reality; artificial intelligence-enabled, self-learning bots; solutions on cognitive services, and other mobility solutions that simplify operations.

During the past five years, in particular, we have been extremely focused on security, as threats at the application, network and data storage levels have increased manifold. We have evolved processes, tapped into combinations of technologies and developed automatic monitoring, logging and alert systems for any intrusion either by malicious users or automated software attacks.

IP is one very critical aspect of how our customers become our partners. We offer them privileged access based on version control so that the IP cannot be accessed through physical workstations on LAN/WAN. Also, we use a secure VPN tunnel to remotely access the solutions of our clients, and we continuously monitor and regularise policy controls for each new development.

We use security monitoring tools and solutions to generate tamper-proof audit control so that the IP cannot be accessed. Additionally, the code release process strictly follows NDAs (non-disclosure agreements) signed with our customers, and encryption of PII data is mandatory for all our offerings.

Q What are your views on the Indian government's first 'Internet of Things Policy'?

The policy is just the tip of the iceberg, but it is evident that the government is very keen and aggressively pursuing the rolling out of IoT infrastructure so that it becomes affordable and benefits 70 per cent of the Indian population living in rural areas.

advertising mantras

Advertising, if done properly, can do wonders for any business. Here are 10 guru mantras to help you understand the impact of advertising and more importantly—how to do it right. Wish you speedy growth this year.

A good advertisement is one which sells the product without drawing attention to itself.

- David Ogilvy

If your advertising goes UNNOTICED everything else is ACADEMIC.

- William Bernbach

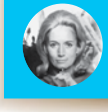
Good advertising does not circulate information.
It penetrates the public mind with desires and belief.

- William Bernbach

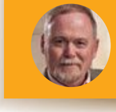
Doing Business Without Advertising is Like Dancing in The Dark. You Know What You're Doing, But Nobody Else Does.

- Stuart H. Britt

The best advertising should make you nervous about what you are not buying.


- Mary Wells Lawrence

Creative without strategy is called ART. Creative with strategy is called ADVERTISING.


- Jef L. Richards

The Business That Considers Itself Immune to The Necessity for Advertising Sooner or Later Finds Itself Immune to Business.

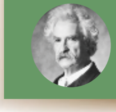
- Derby Brown

If it doesn't sell, it isn't creative.

- David Ogilvy

Branding is what people say about you when you are not in the room.

Many a small thing has been made large by the right kind of advertising.


- Mark Twain

ELECTRONICS FOR YOU

ELECTRONICS BAZAAR

OPEN SOURCE FOR YOU

For more advertising mantras, visit: **http://efy.in/advertising-mantras**

The IoT policy by the government of India

The IoT policy document drafted by the Ministry of Electronics and Information Technology (MeitY) includes the following objectives:

- To create an IoT industry in India with a turnover of US$ 15 billion by 2020. Presumably, India would have a share of 5-6 per cent of the global IoT market.
- To undertake capacity development (human and technology) for IoT-specific skillsets for domestic and international markets.
- To undertake research and development on all the assisting technologies.
- To develop IoT products specific to Indian needs in the domains of agriculture, health, water quality, natural disasters, transportation, security, automobiles, supply chain management, smart cities, automated metering and monitoring of utilities, waste management, as well as oil and gas.

It has been proposed that the framework of the IoT policy will be implemented via a multi-pillar approach. This comprises five vertical pillars, namely, demonstration centres, capacity building and incubation, R&D and innovation, incentives and engagements, human resource development, and two horizontal supports in the form of standards and the governance structure.

To me, the government's initiatives towards IoT are not only exciting but also a strong signal to us to join hands and work to our utmost capacity to make the IoT dream come true.

Q Last of all, where do you see the app market in the next five years? Will it be all-pervasive in India?

The app market is presently in its early phase. My view is that, ultimately, everything is going to be on one or more apps.

There is a huge explosion waiting to happen in the app world, especially in the Indian market. Uber, Haptik, Paytm and Flipkart have just shown us the power of apps. There is much more to come.

Apps will play a key role in connecting all the dots, so that all of us live in a connected way. Right from connected healthcare to smart cities, apps can improve the quality of life not just in urban regions but also in major parts of rural India. App developers are likely to bring about innovations such as predictable farming, smart education for rural children and care for the rural elderly in the near future. All in all, apps are going to be the only media that you will carry in your pocket, 24x7. I think this could happen within the next two years. END

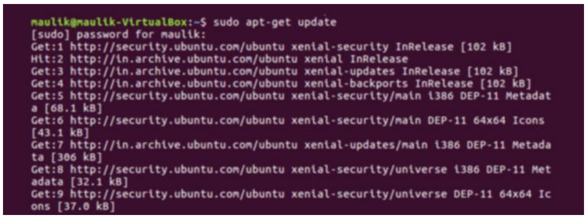
ClamAV: A Free and Open Source Antivirus Tool

ClamAV is a free and open source toolkit to detect malware. It also performs Web/email scanning and provides gateway security for Linux distributions, in particular. Here's a simple guide on how to install and use this tool.

We know that Linux is more secure than Windows, because of which many people think that we don't require antivirus software in Linux. But the fact is that viruses and malware do infiltrate Linux systems too. Though, it is true that the risk is lower compared to Windows. Personally, I haven't found any noxious intruders in my Linux box yet, but we can't say that it will never happen. So it's better to take some precautions to avoid any kind of attack.

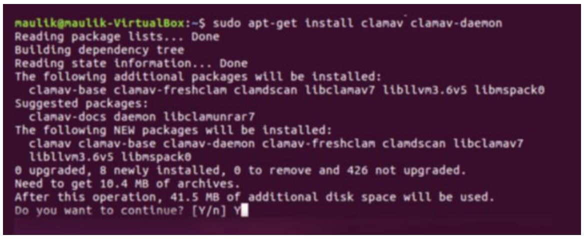
When should ClamAV be used?

- When you have very sensitive data and hence don't want to take any risks, ClamAV will provide an additional level of security.
- Use it when you want to do a system scan without booting into the system, so that viruses and malware do not get activated during the scan.
- When scanning external mails for any malware, since ClamAV is more helpful as a gateway scanner.



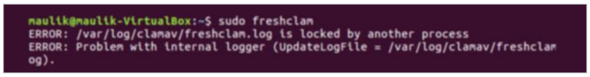
```
maulik@maulik-VirtualBox:~$ sudo apt-get update
[sudo] password for maulik:
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu xenial InRelease
Get:3 http://in.archive.ubuntu.com/ubuntu xenial-updates InRelease [102 kB]
Get:4 http://in.archive.ubuntu.com/ubuntu xenial-backports InRelease [102 kB]
Get:5 http://security.ubuntu.com/ubuntu xenial-security/main i386 DEP-11 Metadat
a [68.1 kB]
Get:6 http://security.ubuntu.com/ubuntu xenial-security/main DEP-11 64x64 Icons
[43.1 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu xenial-updates/main i386 DEP-11 Metada
ta [306 kB]
Get:8 http://security.ubuntu.com/ubuntu xenial-security/universe i386 DEP-11 Met
adata [32.1 kB]
Get:9 http://security.ubuntu.com/ubuntu xenial-security/universe DEP-11 64x64 Ic
ons [37.0 kB]
```

Figure 1: Updating the repository



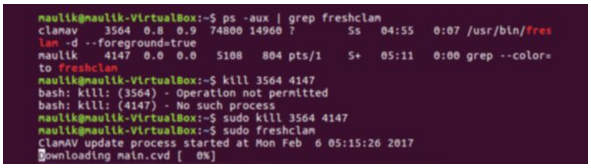
```
maulik@maulik-VirtualBox:~$ sudo apt-get install clamav clamav-daemon
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  clamav-base clamav-freshclam clamdscan libclamav7 libllvm3.6v5 libmspack0
Suggested packages:
  clamav-docs daemon libclamunrar7
The following NEW packages will be installed:
  clamav clamav-base clamav-daemon clamav-freshclam clamdscan libclamav7
  libllvm3.6v5 libmspack0
0 upgraded, 8 newly installed, 0 to remove and 426 not upgraded.
Need to get 10.4 MB of archives.
After this operation, 41.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Figure 2: Installing ClamAV



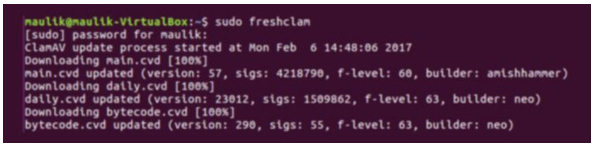
```
maulik@maulik-VirtualBox:~$ sudo freshclam
ERROR: /var/log/clamav/freshclam.log is locked by another process
ERROR: Problem with internal logger (UpdateLogFile = /var/log/clamav/freshclam
og).
```

Figure 3: Error while updating virus database



```
maulik@maulik-VirtualBox:~$ ps -aux | grep freshclam
clamav    3564  0.8  0.9  74800 14960 ?        Ss   04:55   0:07 /usr/bin/fres
lam -d --foreground=true
maulik    4147  0.0  0.0   5108   804 pts/1    S+   05:11   0:00 grep --color=
to freshclam
maulik@maulik-VirtualBox:~$ kill 3564 4147
bash: kill: (3564) - Operation not permitted
bash: kill: (4147) - No such process
maulik@maulik-VirtualBox:~$ sudo kill 3564 4147
maulik@maulik-VirtualBox:~$ sudo freshclam
ClamAV update process started at Mon Feb  6 05:15:26 2017
Downloading main.cvd [  0%]
```

Figure 4: Updating virus database



```
maulik@maulik-VirtualBox:~$ sudo freshclam
[sudo] password for maulik:
ClamAV update process started at Mon Feb  6 14:48:06 2017
Downloading main.cvd [100%]
main.cvd updated (version: 57, sigs: 4218790, f-level: 60, builder: amishhammer)
Downloading daily.cvd [100%]
daily.cvd updated (version: 23012, sigs: 1509862, f-level: 63, builder: neo)
Downloading bytecode.cvd [100%]
bytecode.cvd updated (version: 290, sigs: 55, f-level: 63, builder: neo)
```

Figure 5: Virus database update completed

ClamAV installation

As ClamAV is open source, many third parties have developed different versions of it for different operating systems.

Let's look at how we can install it in Ubuntu. First, update the repository packets lists as follows (this is optional):

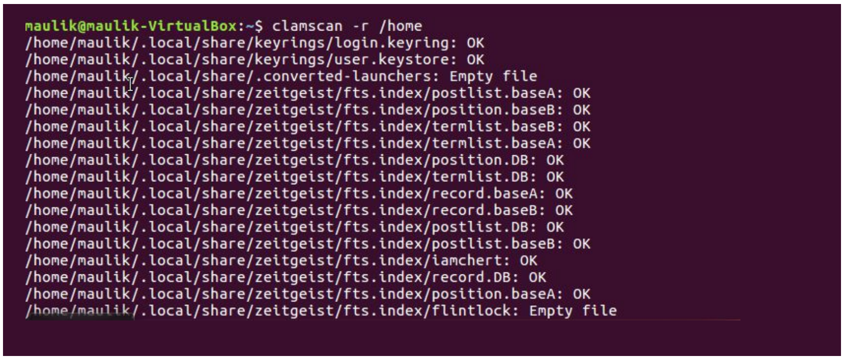
```
sudo apt-get update
```

Issue the command given below to install ClamAV. It will install *clamav-freshclam* also.

```
sudo apt-get install clamav
clamav-daemon
```

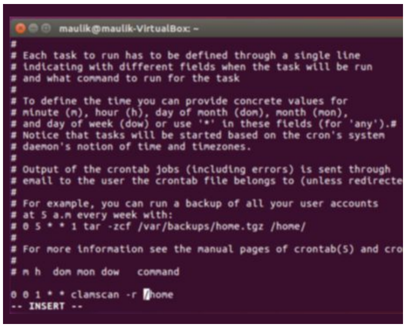
Now ClamAV is installed in our system. The next step is to update the virus definition database. This process is similar to normal updates done when instructing any antivirus software to fetch the latest virus related information. Once we run the command given below, two files -- *main.cvd* and *daily.cvd* --will be downloaded and the virus database will be updated.

```
sudo freeclam
```



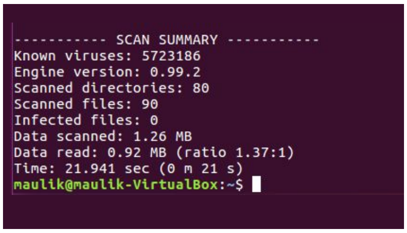
```
maulik@maulik-VirtualBox:~$ clamscan -r /home
/home/maulik/.local/share/keyrings/login.keyring: OK
/home/maulik/.local/share/keyrings/user.keystore: OK
/home/maulik/.local/share/.converted-launchers: Empty file
/home/maulik/.local/share/zeitgeist/fts.index/postlist.baseA: OK
/home/maulik/.local/share/zeitgeist/fts.index/position.baseB: OK
/home/maulik/.local/share/zeitgeist/fts.index/termlist.baseB: OK
/home/maulik/.local/share/zeitgeist/fts.index/termlist.baseA: OK
/home/maulik/.local/share/zeitgeist/fts.index/position.DB: OK
/home/maulik/.local/share/zeitgeist/fts.index/termlist.DB: OK
/home/maulik/.local/share/zeitgeist/fts.index/record.baseA: OK
/home/maulik/.local/share/zeitgeist/fts.index/record.baseB: OK
/home/maulik/.local/share/zeitgeist/fts.index/postlist.DB: OK
/home/maulik/.local/share/zeitgeist/fts.index/postlist.baseB: OK
/home/maulik/.local/share/zeitgeist/fts.index/iamchert: OK
/home/maulik/.local/share/zeitgeist/fts.index/record.DB: OK
/home/maulik/.local/share/zeitgeist/fts.index/position.baseA: OK
/home/maulik/.local/share/zeitgeist/fts.index/flintlock: Empty file
```

Figure 6: Scanning a particular directory



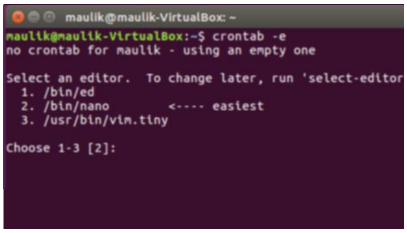
```
maulik@maulik-VirtualBox: ~
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirecte
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cro
#
# m h  dom mon dow   command

0 0 1 * * clamscan -r /home
-- INSERT --
```

Figure 9: Modified *crontab* file



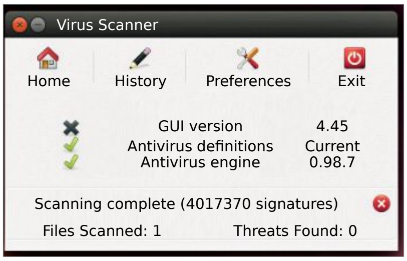
```
----------- SCAN SUMMARY -----------
Known viruses: 5723186
Engine version: 0.99.2
Scanned directories: 80
Scanned files: 90
Infected files: 0
Data scanned: 1.26 MB
Data read: 0.92 MB (ratio 1.37:1)
Time: 21.941 sec (0 m 21 s)
maulik@maulik-VirtualBox:~$
```

Figure 7: Scanning results



```
maulik@maulik-VirtualBox: ~
maulik@maulik-VirtualBox:~$ crontab -e
no crontab for maulik - using an empty one

Select an editor.  To change later, run 'select-editor
  1. /bin/ed
  2. /bin/nano        <---- easiest
  3. /usr/bin/vim.tiny

Choose 1-3 [2]:
```

Figure 8: Scheduling a *cron* job



GUI version 4.45
Antivirus definitions Current
Antivirus engine 0.98.7

Scanning complete (4017370 signatures)

Files Scanned: 1 Threats Found: 0

Figure 10: GUI prompt ClamTK

Figure 3 indicates an error while updating the virus database. This is because after the installation of the ClamAV daemon, the *freeclam* process is already running; so we need to stop it or kill it before running the command again.

The first time, it will take longer to update the database because it is freshly installed.

Now we will scan the */home* directory using ClamAV. Run the command given below to perform the scanning:

```
clamscan -r /home
```

By default, it will update the *freshclam* daemon every hour (24 times). We can change this by using the command given below:

```
sudo dpkg-reconfigure clamav-freshclam
```

To check the version of ClamAV, use the following command:

```
clamdscan -V
```

We can also set a *cron* job for it, so that it will repeatedly scan the mentioned drive/directory as per the given time.

To do that, run the command given below:

```
crontab -e
```

A *crontab* file will be opened and you can append the file in the given (below) link before saving the file:

```
0 0 1 * * clamscan -r /location
```

It will run the ClamAV every first day of the month at midnight (12 a.m.).

Here is the *crontab* format for reference:

```
Minute   Hour   Day of Month  Month
Day of Week         Command
(0-59)     (0-23)     (1-31)    (1-
12 or Jan-Dec) (0-6 or Sun-Sat)
    0         0          1
*                *
clamscan -r  /home
```

Till now, we have seen scanning using CLI; we can do the same thing using a GUI too. Run the command given below to do so:

```
sudo apt-get install ClamTK
```

If you find any difficulties during this step, please refer to the link given below for troubleshooting.

http://askubuntu.com/questions/378558/unable-to-locate-package-while -trying-to-install-packages-with-apt

Similarly, we can install ClamAV for Windows. Refer to the link that follows to download the *.msi* file for the Windows version.

https://www.clamav.net/downloads

There are many third party tools supported by ClamAV, though ClamAV itself does not provide any support for those tools.

You can refer to the official site of ClamAV at *https://www.clamav.net* for more information. END

By: Maulik Parekh

The author has an M. Tech degree in cloud computing from VIT University, Chennai. He can be reached at *maulikparekh2@gmail.com*. Website: *https://www.linkedin.com/in/maulikparekh2*

Prometheus: A Peek at the Popular Monitoring Tool

Prometheus is a leading monitoring solution that has seen its community grow to large numbers. The support for multiple exporters is one of the strongest points of Prometheus, since it can help you get started with specific monitoring requirements quickly.

Prometheus is a leading open source monitoring and alerting tool. It had its origins at SoundCloud and has seen significant adoption since it was announced in 2015. The software was created because of the need to monitor multiple microservices that might be running in your system. Its architecture is modular and comes with several readily available modules called exporters, which help you capture metrics from the most popular software. Prometheus is written in the Go language, and it ships with easily distributed binaries that you can use to get it running as quickly as possible.

This article looks at Prometheus, its architecture and how it can be installed. We then look at an example of monitoring your Linux node with the default support that is available out-of-the-box with Prometheus.

Prometheus' architecture

The architecture of Prometheus, taken from the official documentation, is shown in Figure 1.

The architecture might look complex but we can break it down into modules and their respective roles in the overall system. The key modules are as follows.

The Prometheus server: This is the heart of the system. This server collects the metrics from multiple nodes and stores them locally. The Prometheus server works on the principle of scraping, i.e., invoking the metrics endpoints of the various nodes that it is configured to monitor. It collects these metrics at regular intervals and stores them locally. These metrics are pulled from nodes that run specific exporters (which are modules that extract information and translate it into the Prometheus format, which the server can then ingest). The nodes expose these over the endpoints that the Prometheus server scrapes.

Push gateway: In case the nodes are not exposing an endpoint from which the Prometheus server can collect the metrics, the Prometheus ecosystem has a push gateway. This gateway API is useful for one-off jobs that run, capture the data, transform that data into the Prometheus data format and then push that data into the Prometheus server.

Alert manager: One half of the Prometheus system is about collecting metrics. But of more importance is the ability to define your own alerts on those metrics, so that you can be notified in case of any discrepancies or levels that you might be interested in. This is the job of the alerts manager, which stores not just the alert levels but also can deliver these alerts to you over multiple channels like SMS, email, Slack, etc.

Visualisation: Prometheus comes with its own user interface that you can use to check on the configuration, nodes and graphs. Additionally, it is now compatible with Grafana, a leading open source visualisation application, so that Prometheus data is available for viewing inside Grafana. Prometheus also exposes an API, so in case you are interested in writing your own clients, you can do that too.

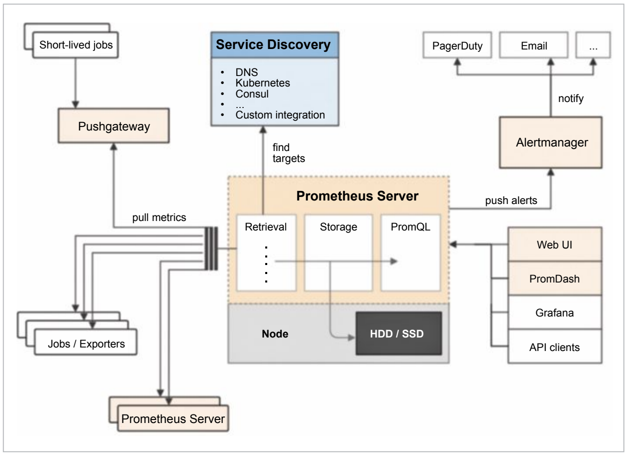


Figure 1: Prometheus' architecture

Installing Prometheus and node exporter

We will now install Prometheus and one of the exporters called the node exporter. The node exporter is an application that runs on a node and can collect various metrics like memory, disk I/O and more. It also exposes an endpoint, which the Prometheus server scrapes at regular intervals and collects the metrics.

Visit the Prometheus downloads page at *https://prometheus.io/download/* and you will see the binaries made available for Prometheus and various other modules like the alerts manager, node exporter and more.

Assuming that you want to install the Prometheus server on a Linux distribution, download the Prometheus server as shown below:

```
wget "https://github.com/prometheus/prometheus/releases/
download/v1.5.0/prometheus-1.5
.0.linux-amd64.tar.gz"
```

Extract the files into a folder and, at the root of that folder, you should see the following key files: *prometheus* and *prometheus.yml*.

Now, let us go and download the node exporter as shown below:

```
wget "https://github.com/prometheus/node_exporter/releases/
download/v0.13.0/node_exporter-0.13.0.darwin-amd64.tar.gz"
```

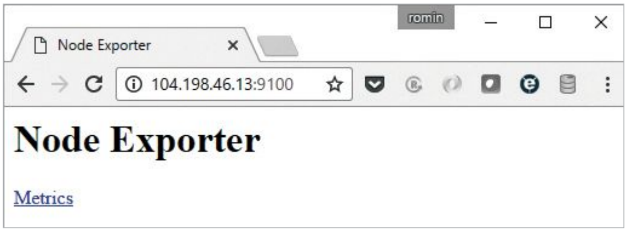
Extract the files into a folder and, at the root of that folder, you should see the following key file: *node_exporter*.

Monitoring your node with Prometheus

In the earlier section on architecture, we saw that the Prometheus server can collect metrics from multiple nodes that need to be monitored. In our example here, we will monitor the Prometheus server itself. We will execute these programs on the same node, for the sake of simplicity.

The first thing that we shall run is the *node_exporter*. The node exporter collects multiple metrics about the node like memory, disk I/O, processes and more. To run the node exporter, go ahead and run the *node_exporter* program that you just downloaded. The output is shown below:

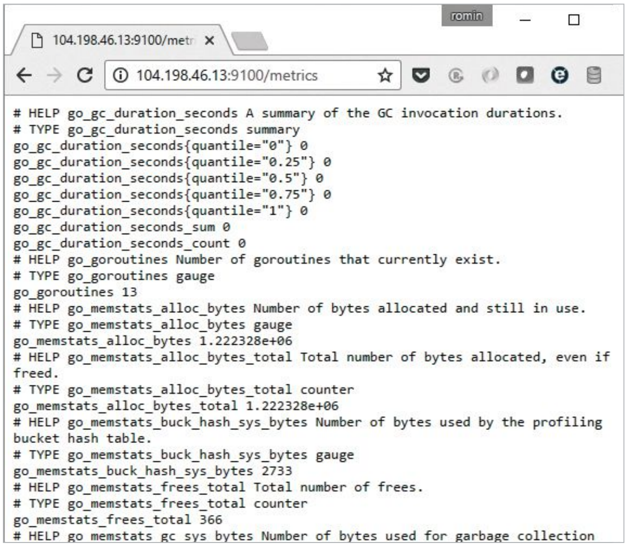
```
$ ./node_exporter  &
[1] 484
romin_irani@promserver:~/Prometheus/node_exporter/node_
exporter-0.13.0.linux-amd64$ INFO[0000] Starting node_
exporter (version=
0.13.0, branch=master, revision=006d1c7922b765f458fe9b92ce646
641bded0f52)  source=node_exporter.go:135
INFO[0000] Build context (go=go1.7.3, user=root@75db7098576a,
date=20161126-13:11:09)  source=node_exporter.go:136
INFO[0000] No directory specified, see --collector.textfile.
directory  source=textfile.go:57
INFO[0000] Enabled collectors:
source=node_exporter.go:155
INFO[0000]  - mdadm
source=node_exporter.go:157
INFO[0000]  - meminfo
source=node_exporter.go:157
INFO[0000]  - vmstat
source=node_exporter.go:157
INFO[0000]  - loadavg
source=node_exporter.go:157
INFO[0000]  - entropy
source=node_exporter.go:157
INFO[0000]  - filefd
source=node_exporter.go:157
INFO[0000]  - netdev
source=node_exporter.go:157
INFO[0000]  - sockstat
source=node_exporter.go:157
INFO[0000]  - textfile
source=node_exporter.go:157
INFO[0000]  - diskstats
source=node_exporter.go:157
INFO[0000]  - netstat
source=node_exporter.go:157
INFO[0000]  - filesystem
source=node_exporter.go:157
INFO[0000]  - hwmon
source=node_exporter.go:157
INFO[0000]  - stat
source=node_exporter.go:157
INFO[0000]  - time
source=node_exporter.go:157
INFO[0000]  - uname
source=node_exporter.go:157
INFO[0000]  - conntrack
source=node_exporter.go:157
INFO[0000] Listening on :9100
source=node_exporter.go:176
```



Node Exporter

Metrics

Figure 2: Node endpoint



```
# HELP go_gc_duration_seconds A summary of the GC invocation durations.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 13
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 1.222328e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if
freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 1.222328e+06
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling
bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 2733
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 366
# HELP go memstats gc sys bytes Number of bytes used for garbage collection
```

Figure 3: Node metrics

You will notice that the node exporter has started to collect various metrics and has exposed a Prometheus metrics data compatible endpoint on port 9100. If we visit the endpoint *http://<your-node-ip>:9100*, you will see the node endpoint as shown in Figure 2.

Click on the *Metrics* link and it will display multiple metrics that are captured. A sample screenshot is shown in Figure 3.

Now that the node exporter is working fine, let us start the Prometheus server. Before we start the Prometheus server, we need to identify the nodes from which it will scrape the node metrics.

Go to the folder into which you extracted the core Prometheus server files, i.e., *prometheus* and *prometheus.yml*. The YAML file is the key configuration file and will define multiple targets that the Prometheus server needs to scrape. In addition to targets, it can also have multiple other configuration entries like alerts, default time intervals, etc.

Since we are only interested in monitoring one node, which is running locally for the moment, the *prometheus.yml* file is shown below:

```
#my global config
global:
  scrape_interval:     15s
  evaluation_interval: 15s
scrape_configs:
  - job_name: 'node'
    static_configs:
      - targets: ['localhost:9100']
```

Now, let us launch the Prometheus server as shown below:

```
$ ./prometheus
INFO[0000] Starting prometheus (version=1.5.0, branch=master,
revision=d840f2c400629a846b210cf58d65b9fbae0f1d5c)
source=main.go:75
INFO[0000] Build context (go=go1.7.4, user=root@a04ed5b536e3,
date=20170123-13:56:24)  source=main.go:76
INFO[0000] Loading configuration file prometheus.yml
source=main.go:248
INFO[0000] Loading series map and head chunks...
source=storage.go:373
INFO[0001] 1203 series loaded.
source=storage.go:378
INFO[0001] Starting target manager...
source=targetmanager.go:61
INFO[0001] Listening on :9090
source=web.go:259
```

You can see that the Prometheus server is listening on port 9090, and we can use that information to take a peek into its default Web user interface (UI).

Visit the endpoint on port 9090 as shown in Figure 4.

You can now enter an expression for one of the metrics that you want to take a look at — for example, the node CPU metrics,

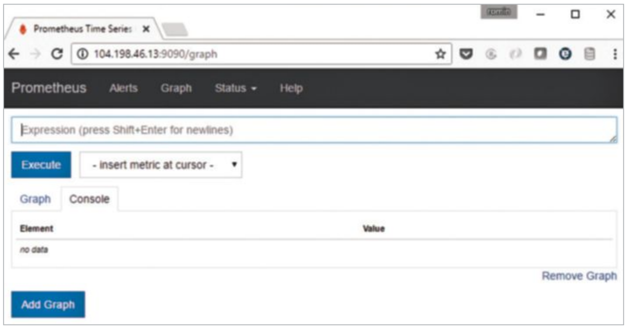
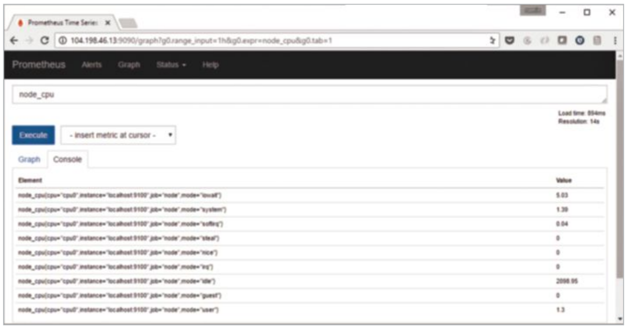


Figure 4: Prometheus server UI



node_cpu

Element	Value
node_cpu{cpu="cpu0",instance="localhost:9100",job="node",mode="iowait"}	5.03
node_cpu{cpu="cpu0",instance="localhost:9100",job="node",mode="system"}	1.39
node_cpu{cpu="cpu0",instance="localhost:9100",job="node",mode="softirq"}	0.04
node_cpu{cpu="cpu0",instance="localhost:9100",job="node",mode="steal"}	0
node_cpu{cpu="cpu0",instance="localhost:9100",job="node",mode="nice"}	0
node_cpu{cpu="cpu0",instance="localhost:9100",job="node",mode="irq"}	0
node_cpu{cpu="cpu0",instance="localhost:9100",job="node",mode="idle"}	2098.95
node_cpu{cpu="cpu0",instance="localhost:9100",job="node",mode="guest"}	0
node_cpu{cpu="cpu0",instance="localhost:9100",job="node",mode="user"}	1.3

Figure 5: Prometheus server graphs

Lots to learn, lots to do

"Prioritize this. Wait! Now work on that"
Soul-sucking job, but it pays for the flat
Boss is a jerk, team full of fakes
Deep down I still know I've got what it takes
To break away, surge ahead, build something true
so my country and mom can be proud of it too

Loonycorn

Our Content:

- The Complete Machine Learning Bundle
  10 courses | 63 hours | $39

- The Complete Computer Science Bundle
  8 courses | 78 hours | $39

- The Big Data Bundle
  9 courses | 64 hours | $45

- The Complete Web Programming Bundle
  8 courses | 61 hours | $41

- The Complete Finance & Economics Bundle
  9 courses | 56 hours | $49

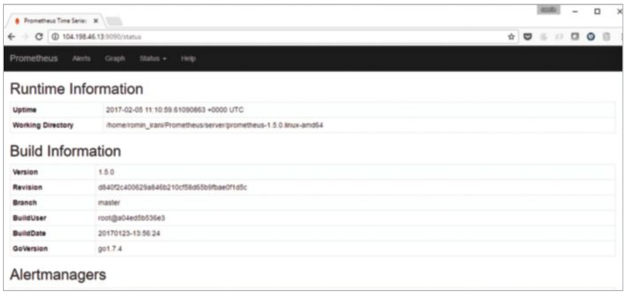
- The Scientific Essentials Bundle
  7 courses | 41 hours | $35

- ~15 courses on Pluralsight
  ~70 on StackSocial
  ~60 on Udemy

About Us:

- 50,000+ students - most of them happy:-)

- <10 team - all of us happy:-)
  ex-Google | Stanford | IIM-Ahmedabad | INSEAD



Runtime Information

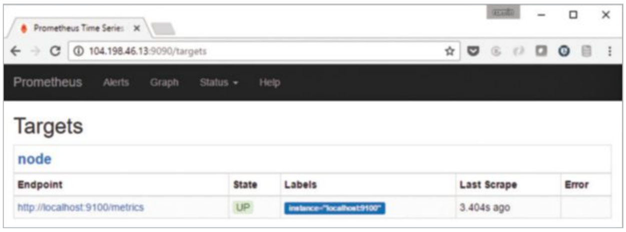
Uptime	2017-02-05 11:10:59.61090863 +0000 UTC
Working Directory	/home/romin_irani/Prometheus/server/prometheus-1.5.0.linux-amd64

Build Information

Version	1.5.0
Revision	d840f2c400629a846b210cf58d65b9fbae0f1d5c
Branch	master
BuildUser	root@a04ed5b536e3
BuildDate	20170123-13:56:24
GoVersion	go1.7.4

Alertmanagers

Figure 6: Prometheus server configuration



Targets

node

Endpoint	State	Labels	Last Scrape	Error
http://localhost:9100/metrics	UP	instance="localhost:9100"	3.404s ago	

Figure 7: Prometheus server targets configuration

which is shown in Figure 5. Just select it in the expressions list and click on the *Execute* button, as shown in the figure.

There is useful information in the *Status* menu option also. You can view your configuration, rules, alerts and targets from there. The default information is shown in Figure 6.

You can click on the *Targets* link in the *Status* main menu to see the targets that you have configured. Since we have configured only one target, that is what we see (Figure 7).

You can view the nodes from the Prometheus UI by visiting */consoles/node.html endpoint* as shown in Figure 8.

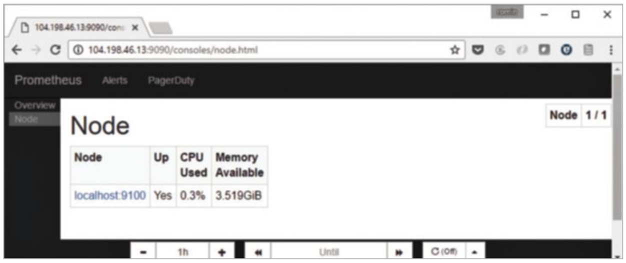
You can now click on the *Node* link to see more metrics about the node.

This completes the steps on validating your basic Prometheus set-up. We have only touched upon the surface of this topic. Your next steps should be to assess the several exporters that are available and see which ones address your monitoring requirements, along with any alerts that you would like to set up for your environment.

Cloud Native Computing Foundation (CNCF) and Prometheus

The Cloud Native Computing Foundation (CNCF) is, according to its website, a non-profit organisation committed to advancing the development of cloud native applications and services by creating a new set of common container technologies guided by technical merit and end user value, and inspired by Internet-scale computing.

The first project to be accepted by this foundation was Kubernetes, which is the leading open source solution for container orchestration. Prometheus has been accepted as the second project by this foundation, and that speaks volumes about its functionality and its acceptance as a standard in monitoring applications and microservices.

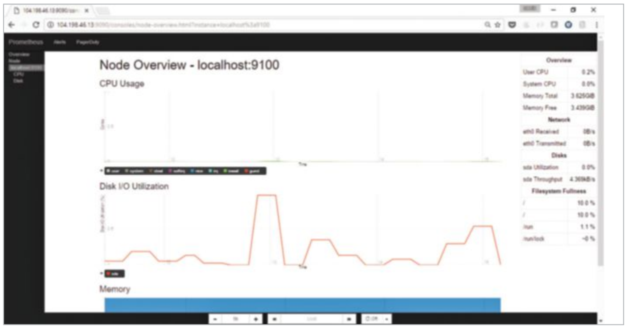


Node

Node	1 / 1

Node	Up	CPU Used	Memory Available
localhost:9100	Yes	0.3%	3.519GiB

Figure 8: Prometheus server - node list



Node Overview - localhost:9100

Overview	
User CPU	0.2%
System CPU	0.0%
Memory Total	3.625GiB
Memory Free	3.439GiB
Network	
eth0 Received	0B/s
eth0 Transmitted	0B/s
Disks	
sda Utilization	0.0%
sda Throughput	4.369kB/s
Filesystem Fullness	
/	10.0 %
/	10.0 %
/run	1.1 %
/run/lock	~0 %

Figure 9: Node metrics via the visualisation dashboard

Prometheus integrates with CNCF's first hosted project, Kubernetes, to support service discovery and monitoring of dynamically scheduled services. Kubernetes also supports Prometheus natively.

Contributing to Prometheus

One of the key reasons for the growth of Prometheus has been the contributions it has received from the community. But often, it is a challenge to understand how you can get started with contributing to some of its core modules. In an article published at NewStack, titled 'Contributing to Prometheus: An Open Source Tutorial', the writer takes up the architecture of the alerts manager and breaks down that module to help us understand how it works and the potential ways in which we can contribute.

Having been accepted by the Cloud Native Computing Foundation, it would help tremendously to contribute to this project, since the visibility of your contribution will be very high. END

References

[1] Prometheus Home Page: *https://prometheus.io/*
[2] Prometheus Downloads: *https://prometheus.io/download/*
[3] Prometheus Exporters: *https://prometheus.io/docs/instrumenting/exporters/*
[4] Contributing to Prometheus: An Open Source Tutorial : *http://thenewstack.io/contributing-prometheus-history-alertmanager/*
[5] Prometheus and CNCF: *https://goo.gl/VRBfTA*

By: Romin Irani

The author has over 20 years' experience in the software industry. His passion is to read and write about technology, as well as to teach it. He blogs at *www.rominirani.com*.

Use Content Delivery Networks to Speed Up Your Website

A content delivery network (CDN) has multiple servers, which are geographically distributed for the easy delivery of Web content. A CDN enhances the performance and speed of delivery of such content.

One of the significant ways of enhancing website performance is to implement a CDN to serve compiled assets in Rails.

The problem

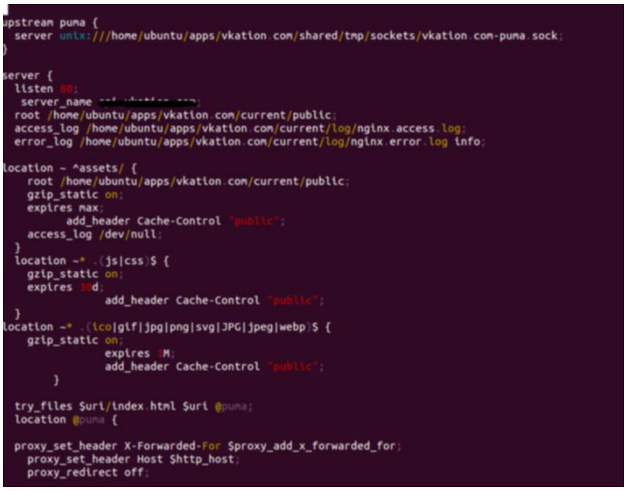
Though the addition of an asset pipeline decreases the number of assets served and the file size, yet the speed at which the contents are transmitted to the end user is slow. Distance also plays a role in the speed at which data can be delivered. Because of slow connections, the user's patience decreases, and so does your ability to effectively engage them.

The solution: Content delivery networks (CDNs)

CDNs are networks of servers that host your content so that when you make a request, the request is served from the server closest to you. The use of a CDN also reduces the number of requests hitting your application server (Apache/ nginx).

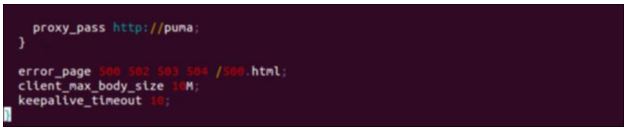
Nginx default site configuration for Ruby on Rails is given below:

```
upstream puma {
  server unix:///home/ubuntu/apps/example.com/shared/tmp/
sockets/vkation.com-puma.sock;
}

server {
  listen 80;
   server_name example.com;
  root /home/ubuntu/apps/example.com/current/public;
  access_log /home/ubuntu/apps/example.com/current/log/nginx.
access.log;
  error_log /home/ubuntu/apps/example.com/current/log/nginx.
error.log info;

location ~ ^assets/ {
    root /home/ubuntu/apps/example.com/current/public;
    gzip_static on;
    expires max;
          add_header Cache-Control "public";
    access_log /dev/null;
  }
  location ~* .(js|css)$ {
    gzip_static on;
    expires 30d;
                add_header Cache-Control "public";
  }
location ~* .(ico|gif|jpg|png|svg|JPG|jpeg|webp)$ {
    gzip_static on;
                expires 1M;
                add_header Cache-Control "public";
        }

 try_files $uri/index.html $uri @puma;
  location @puma {

  proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_
for;
    proxy_set_header Host $http_host;
    proxy_redirect off;

    proxy_pass http://puma;
  }
```



```
upstream puma {
  server unix:///home/ubuntu/apps/vkation.com/shared/tmp/sockets/vkation.com-puma.sock;
}

server {
  listen 80;
   server_name [redacted];
  root /home/ubuntu/apps/vkation.com/current/public;
  access_log /home/ubuntu/apps/vkation.com/current/log/nginx.access.log;
  error_log /home/ubuntu/apps/vkation.com/current/log/nginx.error.log info;

location ~ ^assets/ {
    root /home/ubuntu/apps/vkation.com/current/public;
    gzip_static on;
    expires max;
          add_header Cache-Control "public";
    access_log /dev/null;
  }
  location ~* .(js|css)$ {
    gzip_static on;
    expires 30d;
                add_header Cache-Control "public";
  }
location ~* .(ico|gif|jpg|png|svg|JPG|jpeg|webp)$ {
    gzip_static on;
                expires 1M;
                add_header Cache-Control "public";
        }

  try_files $uri/index.html $uri @puma;
  location @puma {

  proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    proxy_redirect off;
```

Figure 1: Nginx configuration



```
   proxy_pass http://puma;
  }

  error_page 500 502 503 504 /500.html;
  client_max_body_size 10M;
  keepalive_timeout 10;
}
```

Figure 2: Nginx configuration (contd)

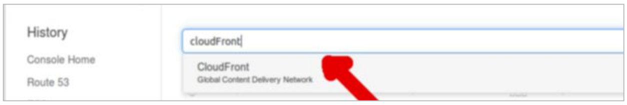


Figure 3: Select cloudfront

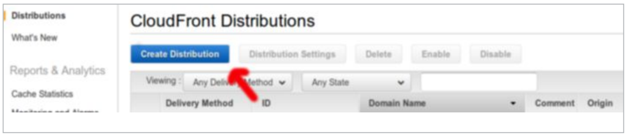
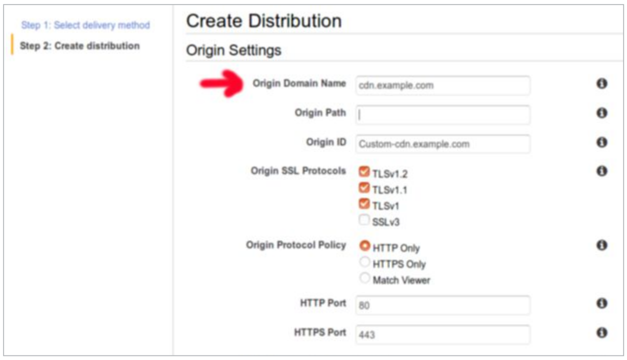


Figure 4: Create a distribution



Create Distribution

Origin Settings

Origin Domain Name: cdn.example.com

Origin Path:

Origin ID: Custom-cdn.example.com

Origin SSL Protocols: ☑ TLSv1.2 ☑ TLSv1.1 ☑ TLSv1 □ SSLv3

Origin Protocol Policy: ● HTTP Only ○ HTTPS Only ○ Match Viewer

HTTP Port: 80

HTTPS Port: 443

Figure 5: Enter the domain name

```
  error_page 500 502 503 504 /500.html;
  client_max_body_size 10M;
  keepalive_timeout 10;
}
```

CDN configuration

In this configuration, the CDN pulls assets from your Web server and caches them. All subsequent requests to that asset will be served straight from the CDN.

1. Login to *AWS* console and select *Cloudfront* as shown in Figure 3.
2. Click *Create Distribution* as shown in Figure 4.
3. Enter the domain name where your assets are currently located as shown in Figure 5.
4. Customise the object caching. Minimum TTL is set to one day.
5. Enter the alternate domain name, if any, and keep the rest as defaults.
6. Make note of the Cloudfront distribution URL.
7. Enter the Cloudfront distribution URL in the Rails *asset_host* to change the host URL of the assets, as follows:

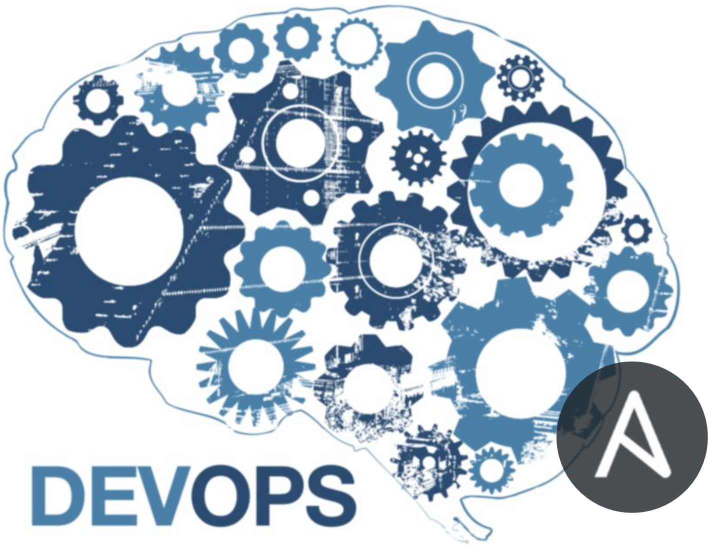
```
# config/environments/production.rb
config.action_controller.asset_host = "XXXX.cloudfront.net"
```

With this, site performance will be increased and the number of requests hitting the application server will also be reduced. END

By: Jayashree N.

The author is a cloud specialist working with Fcoos Technologies. She has expertise in providing technical solutions related to cloud services, particularly on AWS, Azure and OVH, apart from multiple other platforms.

The DevOps Series
An Introduction to Ansible



With this article, we begin a new series on DevOps, starting out with Ansible, which helps you to build a strong foundation. As the Ansible website proclaims, proudly, “Deploy apps. Manage systems. Crush complexity.”

Ansible is an IT automation tool that is used for provisioning, configuration, deployment and managing infrastructure. The project was first released in 2012, and is written in Python. The main objective of the tool is to be simple and easy to use. It is based on an agent-less (push-based) architecture, and the playbooks are written in plain English. It also supports pull-based deployments Ansible has had pull support since 2012 and uses SSH to execute commands on remote machines. It is available under the GNU General Public License.

Installation

You can install Ansible using your GNU/Linux distribution package manager.

On Fedora, you can use Yum to install Ansible, as follows:

```
$ sudo yum install ansible
```

If you are using RHEL or CentOS, install the epel-release, and then use the Yum command to install Ansible.

On Ubuntu, you need to add the *ppa* repository before installing the tool, as shown below:

```
$ sudo apt-get install software-properties-common
$ sudo apt-add-repository ppa:ansible/ansible
```

```
$ sudo apt-get update
$ sudo apt-get install ansible
```

The Ansible documentation encourages Debian users to access the Ubuntu repository to obtain Ansible. You need to add the following line to */etc/apt/sources.list:*

```
deb http://ppa.launchpad.net/ansible/ansible/ubuntu trusty main
```

You can then install the software using the following commands:

```
$ sudo apt-get update
$ sudo apt-get install ansible
```

The Parabola GNU/Linux-libre distribution is a derivative of Arch Linux, without the binary blobs. You can install Ansible using the *pacman* utility:

```
$ pacman -S ansible
```

The latest Ansible version 2.2 (as of date) is what we will use in this article. Ansible is also available for BSD variants, Mac OS X, and Windows. You are encouraged to refer to the Ansible documentation for more information.

Virtualisation

Ansible can be used to provision new machines and also configure them. Instead of using bare metal machines, you can create multiple virtual machines (VMs) on your system. Lots of free and open source software (FOSS) virtualisation software is available.

QEMU is a machine emulator and virtualiser. It can also use host CPU support to run guest VMs for better performance. It is written by Fabrice Bellard, and released under the GNU General Public License (GPL). You can install it on Parabola GNU/Linux-libre, using the following command:

```
$ sudo pacman -S qemu
```

KVM or kernel-based virtual machine has direct support in the Linux kernel. It requires hardware support to be able to run guest operating systems. It is written in C, and is released under the GNU General Public License.

You need to check if your hardware first supports KVM. The *'lscpu'* command will show an entry for *'Virtualization'* if there is hardware support. For example:

```
$ lscpu

Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Thread(s) per core:    2
Core(s) per socket:    2
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 78
Model name:            Intel(R) Core(TM) i5-6200U CPU @
2.30GHz
Stepping:              3
CPU MHz:               2275.341
CPU max MHz:           2800.0000
CPU min MHz:           400.0000
BogoMIPS:              4801.00
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              256K
L3 cache:              3072K
NUMA node0 CPU(s):     0-3

Flags:                 fpu vme de pse tsc msr pae mce cx8
apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr
sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art arch_perfmon pebs bts rep_good nopl xtopology
nonstop_tsc aperfmperf eagerfpu pni pclmulqdq dtes64 monitor
ds_cpl vmx est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1
sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx
f16c rdrand lahf_lm abm 3dnowprefetch epb intel_pt tpr_shadow
vnmi flexpriority ept vpid fsgsbase tsc_adjust bmi1 avx2 smep
bmi2 erms invpcid mpx rdseed adx smap clflushopt xsaveopt
xsavec xgetbv1 xsaves dtherm ida arat pln pts hwp hwp_notify
hwp_act_window hwp_epp
```

You can also check the */proc/cpuinfo* output as shown below:

```
$ grep -E "(vmx|svm)" --color=always /proc/cpuinfo

flags           : fpu vme de pse tsc msr pae mce cx8 apic sep
mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2
ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc art
arch_perfmon pebs bts rep_good nopl xtopology nonstop_tsc
aperfmperf eagerfpu pni pclmulqdq dtes64 monitor ds_cpl vmx
est tm2 ssse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2
x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c
rdrand lahf_lm abm 3dnowprefetch epb intel_pt tpr_shadow vnmi
flexpriority ept vpid fsgsbase tsc_adjust bmi1 avx2 smep bmi2
erms invpcid mpx rdseed adx smap clflushopt xsaveopt xsavec
xgetbv1 xsaves dtherm ida arat pln pts hwp hwp_notify hwp_
act_window hwp_epp
```

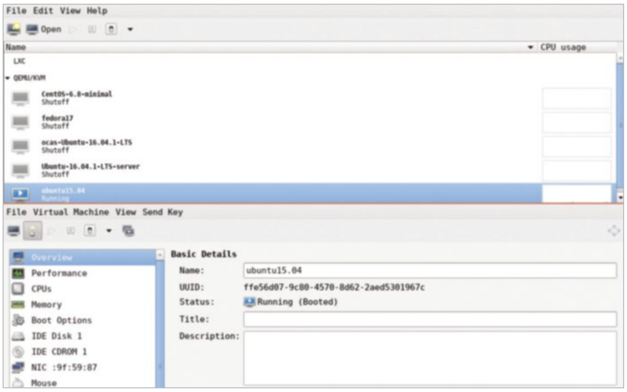
The Libvirt project provides APIs to manage guest machines on KVM, QEMU and other virtualisation software. It is written in C, and is released under the GNU Lesser GPL. The virtual machine manager (VMM) provides a graphical user interface for managing the guest VMs and is written in Python.

You can install all this software on Parabola GNU/Linux-Libre using the following command:

```
$ sudo pacman -S libvirt virt-manager
```

A screenshot of VMM is provided in Figure 1.

Check your distribution documentation to install the appropriate virtualisation software packages.



Name: ubuntu15.04
UUID: ffe56d07-9c80-4570-8d62-2aed5301967c
Status: Running (Booted)
Title:
Description:

Figure 1: Virtual Machine Manager

You can use the VMM to create a new virtual machine, and install a GNU/Linux distribution using a *.iso* image. You can specify RAM, disk size and follow the installation steps for your particular distro. You can also import an existing *.qcow2* disk image to use it as a virtual machine.

Ansible with libvirt-VM

The version of Ansible used for this article is given below:

```
$ ansible --version
ansible 2.2.1.0
  config file = /etc/ansible/ansible.cfg
  configured module search path = Default w/o overrides
```

If you have the *sshd* daemon running on your local machine, you can use Ansible to test it. For example, a ping test on the localhost is shown below:

```
$ ansible localhost -m ping
localhost | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

You can also check how long the system has been up and running using the following commands:

```
$ ansible localhost -a uptime
localhost | SUCCESS | rc=0 >>

 11:00:20 up  4:09,  0 users,  load average: 0.18, 0.14, 0.11
```

You can execute a shell command on the remote machine (localhost, in this case) as illustrated below:

```
$  ansible localhost -a "date"
localhost | SUCCESS | rc=0 >>
Sun Feb  5 11:24:53 IST 2017
```

The 'setup' command provides details of the remote target machine. A snippet output is provided below:

```
$  ansible localhost -m setup

localhost | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "192.168.10.1",
            "192.168.5.6"
        ],
        "ansible_all_ipv6_addresses": [
            "fe90::fc24:ff:feb9:cb61",
            "ff80::5846:fac1:6afc:2e30"
        ],
        "ansible_architecture": "x86_64",
        "ansible_bios_date": "06/12/2016",
        "ansible_bios_version": "R00ET45W (1.20 )",
        "ansible_cmdline": {
            "BOOT_IMAGE": "/vmlinuz-linux-libre",
            "cryptdevice": "/dev/sda1:cryptroot",
            "quiet": true,
            "root": "/dev/mapper/cryptroot",
            "rw": true
        },
        ....
```

An Ubuntu 15.04 instance with VMM is used in the following examples with Ansible. The IP address of the instance is added to */etc/hosts*:

```
192.168.122.250 ubuntu
```

The */etc/ansible/hosts* file contains the following:

```
ubuntu
```

You can now do a ping test from the host to the Ubuntu VM using the following command sequence for the user 'xetex':

```
$  ansible ubuntu -m ping -u xetex --ask-pass
SSH password:
ubuntu | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

To avoid prompting for the password, you can add the localhost public SSH key to the VM, as follows:

```
$  ssh-copy-id -i ~/.ssh/id_rsa.pub xetex@ubuntu

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/home/user/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new
key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed
-- if you are prompted now it is to install the new keys
xetex@ubuntu's password:

Number of key(s) added: 1
```

Now try logging into the machine, with *ssh xetex@ubuntu* and check to make sure that only the key(s) you wanted were added.

Continued to page 44...

How to Use the Network Security Toolkit

Network Security Toolkit (NST) is an analysis and validation tool which can be used on enterprise virtual servers that host virtual machines. Get acquainted with NST in this article.

NST provides a security toolkit for network administrators. It comes with a complete set of open source network security tools and an advanced Web user interface (WUI).

System requirements

The Network Security Toolkit is designed to provide many tools that run entirely in the RAM. Hence, it requires a large amount of RAM. Table 1 lists the minimum system requirements.

Downloading and burning the ISO image

We can read how to download NST Linux and burn the ISO image from the NST instructional exercise. When it is done, restart the PC and begin the installation. NST installs easily on Linux systems.

Logging in

When the installation is done, NST Linux creates a client named 'NST user'. You can log in with this user name and no password.

NST is integrated with different types of tools. You can see them when you access applications from the *Activities* menu on the top left.

NST Web user interface (WUI)

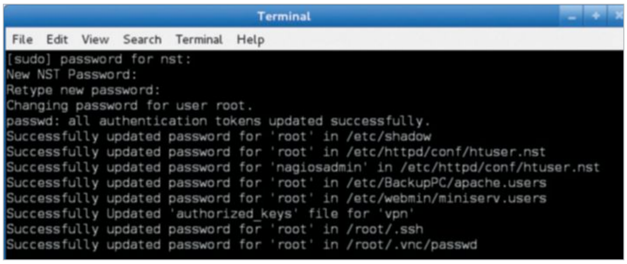
The tools that appear on the desktop are just a small part of the NST armoury. The originality of NST Linux is the NST WUI, which is the control panel or system management tool for all that you have to do with NST. This feature can't be accessed unless you fix a password for the current client. To fix or change the secret password, double-click on the 'Set the NST system passwords'

Table 1

Component	Minimum requirements	Recommended requirements	Notes
CPU	Celeron	i686	It won't keep running on an Intel 386, Intel 486, or Intel Pentium class CPU. It is known to chip away at the Intel Celeron (466MHz) or higher, Intel Pentium II (266MHz) or higher, AMD Athlon, AMD Duron, AMD Athlon XP, AMD Opteron, and AMD Athlon 64.
RAM	128MB	256MB	The base measure of 128MB is sufficient for basic applications. If you want to run X or any serious set of applications, you will need at least 256MB of RAM.
Motherboard	-	-	Motherboard should be supported by Fedora 4 Operating System.
Ethernet	0	2	You could utilise the Network Security Toolkit without an Ethernet card introduced. In any case, it wouldn't be of much use for networks. No less than two Ethernet ports are unequivocally prescribed, allowing one to be utilised for general access to the NST and the other(s) to go about as a test to monitor network traffic.
CDROM	24X	52Xw	It would most likely work on a 4X CDROM, yet the 'get to' time would only be moderate. The NST will boot from a remotely associated USB CD ROM drive if your BIOS permits it.

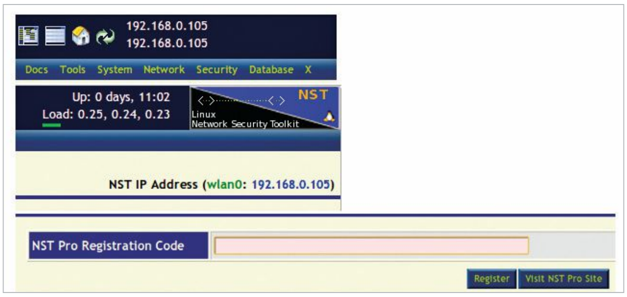
Figure 1: NST login

Figure 2: NST tools view



```
[sudo] password for nst:
New NST Password:
Retype new password:
Changing password for user root.
passwd: all authentication tokens updated successfully.
Successfully updated password for 'root' in /etc/shadow
Successfully updated password for 'root' in /etc/httpd/conf/htuser.nst
Successfully updated password for 'nagiosadmin' in /etc/httpd/conf/htuser.nst
Successfully updated password for 'root' in /etc/BackupPC/apache.users
Successfully updated password for 'root' in /etc/webmin/miniserv.users
Successfully Updated 'authorized_keys' file for 'vpn'
Successfully updated password for 'root' in /root/.ssh
Successfully updated password for 'root' in /root/.vnc/passwd
```

Figure 3: Accessing NST WUI

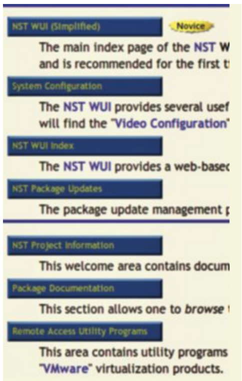


Up: 0 days, 11:02
Load: 0.25, 0.24, 0.23

NST IP Address (wlan0: 192.168.0.105)

NST Pro Registration Code

Figure 4: NST *Start* page



The main index page of the NST W
and is recommended for the first t

The NST WUI provides several usef
will find the "Video Configuration"

The NST WUI provides a web-basec

The package update management p

This welcome area contains docum

This section allows one to *browse*

This area contains utility programs
"VMware" virtualization products.

Figure 5: NST WUI menu

symbol. You will be prompted for a new password, or to change the last secret password that you had.

Once logged in, you can get to NST WUI. Open the Mozilla Firefox browser, and type *http://192.0.0.1/untwist* in the address bar. You will be asked for a login password.

Since it is a Web tool, you can also get it through another machine. The difference is that you need to use the HTTPS protocol to get to NST WUI via the Web.

The NST *Start* page

On the NST WUI page you will see the following:

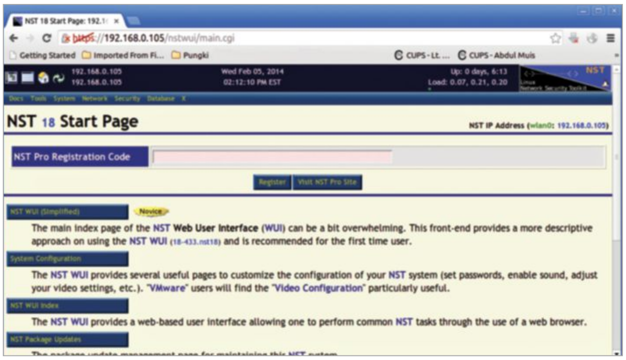
- A menu on the upper left
- The NST IP address and to what extent it has been running
- The NST Pro Registration Code screen

NST Linux: Setting up BandwidthD

BandwidthD is a network traffic test that shows an outline of system use. To enable this feature, go to the menu and follow *Network > Monitors > BandwidthD UI.*

Next, pick the network interface that we need to monitor, arrange the parameter and its subnet. Click the *Start BandwidthD* button.

NST provides two different interfaces for verifying BandwidthD. The first is the important BandwidthD



NST 18 Start Page: 192.1

https://192.168.0.105/nstwui/main.cgi

Getting Started | Imported From Fi... | Pungki | CUPS - Lt.... | CUPS - Abdul Muis

192.168.0.105
192.168.0.105

Wed Feb 05, 2014
02:12:10 PM EST

Up: 0 days, 6:13
Load: 0.07, 0.21, 0.20

NST 18 Start Page NST IP Address (wlan0: 192.168.0.105)

NST Pro Registration Code

Register | Visit NST Pro Site

NST WUI (Simplified) Novice

The main index page of the NST **Web User Interface** (WUI) can be a bit overwhelming. This front-end provides a more descriptive approach on using the NST WUI (18-433.nst18) and is recommended for the first time user.

System Configuration

The NST WUI provides several useful pages to customize the configuration of your NST system (set passwords, enable sound, adjust your video settings, etc.). "VMware" users will find the "Video Configuration" particularly useful.

NST WUI Index

The NST WUI provides a web-based user interface allowing one to perform common NST tasks through the use of a web browser.

NST Package Updates

The package update management page for maintaining this NST system.

Docs Tools System Network Security Database X
NST
Linux Network Security Toolkit

Figure 6: The NST WUI landing page

interface (see Figure 8).

The second is the NST WUI BandwidthD interface, for real-time monitoring with a graph (Figure 9).

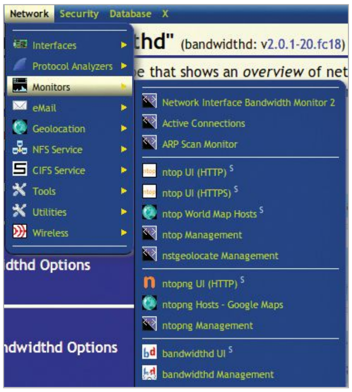


Figure 7: BandwidthD UI

Monitor CPU utilisation

When one or more activities are running simultaneously, we may want to know about the CPU utilisation. NST provides us a tool to do that. To use this tool, go to *System* > *Processes* > *CPU Usage Monitor.* You need to wait for a few seconds to get the graph.

SSH to the server

When you need to carry out a remote activity through the shell, you can do it via the Web. NST Linux provides this function. Simply go to *System* > *Control Management* > *Run command*

At that point, you will have a SSH client on the Web.

Launch the X Window application

With this feature, you can dispatch the X Window application without a remote to the server. The application's graphical acquaintance will be redirected by the X Server on your client PC. In any case, before doing this, you have to ensure that the X Server on your PC acknowledges the TCP connection.

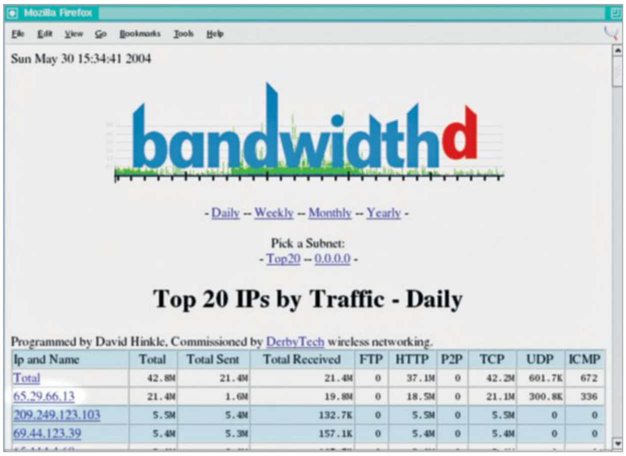
Here is an example. I am using Zorin Linux 7, which is based on Ubuntu, as a client. Here are the steps to be followed.

1. Enable XDMCP as shown below:

```
$ sudo vi /etc/lightdm/lightdm.conf
```

Add the following lines:

```
xserver-allow-tcp=true
[XDMCPServer]
enabled=true
```



Mozilla Firefox

Sun May 30 15:34:41 2004

- Daily -- Weekly -- Monthly -- Yearly -

Pick a Subnet:
- Top20 -- 0.0.0.0 -

Top 20 IPs by Traffic - Daily

Programmed by David Hinkle, Commissioned by DerbyTech wireless networking.

Ip and Name	Total	Total Sent	Total Received	FTP	HTTP	P2P	TCP	UDP	ICMP
Total	42.8M	21.4M	21.4M	0	37.1M	0	42.2M	601.7K	672
65.29.66.13	21.4M	1.6M	19.8M	0	18.5M	0	21.1M	300.8K	336
209.249.123.103	5.5M	5.4M	132.7K	0	5.5M	0	5.5M	0	0
69.44.123.39	5.4M	5.3M	157.1K	0	5.4M	0	5.4M	0	0

Figure 8: BandwidthD interface

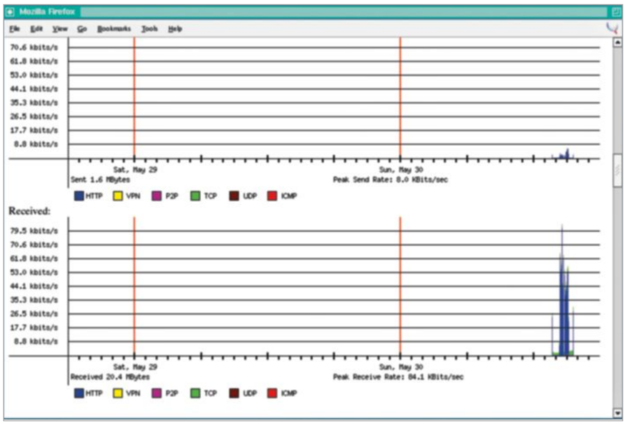
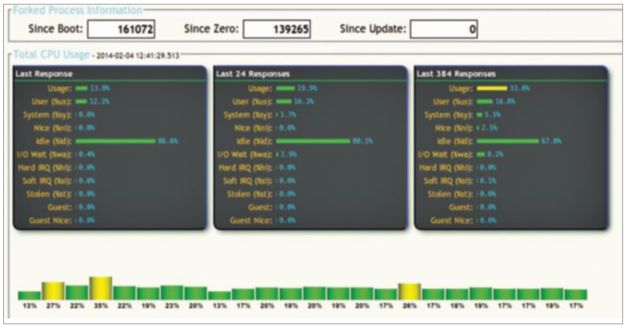


Figure 9: BandwidthD interface with graph



Forked Process Information

Since Boot: 161072 Since Zero: 139265 Since Update: 0

Total CPU Usage - 2014-02-04 12:41:29.513

Figure 10: CPU utilisation graph

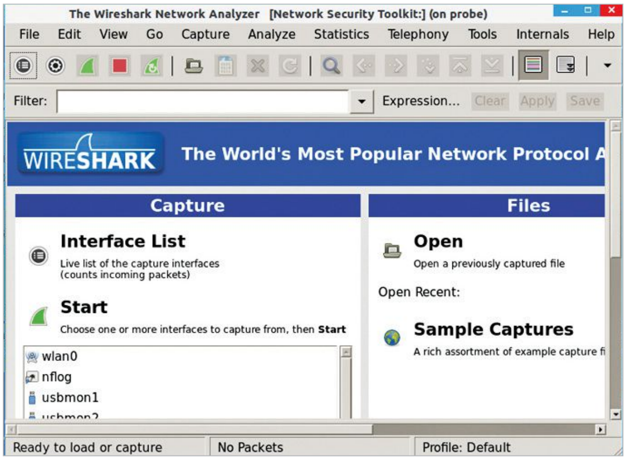
2. Restart *lightdm*, as follows:

```
$ sudo restart lightdm
```

Note: This command will restart your X Window. Every single open application will be shut.

3. Ensure that port 6000 is tuning in. Run *netstat* to check it, as shown below:

```
$ netstat -an | grep -F 6000
```


The Wireshark Network Analyzer [Network Security Toolkit:] (on probe)
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save
WIRESHARK
The World's Most Popular Network Protocol A
Capture
Files
Interface List
Start
Open
Open Recent:
Sample Captures
wlan0
nflog
usbmon1
usbmon2
Ready to load or capture
No Packets
Profile: Default

Live list of the capture interfaces (counts incoming packets)

Choose one or more interfaces to capture from, then Start

Open a previously captured file

A rich assortment of example capture fi

Figure 11: Wireshark interface

4. Allow your computer to accept the X Server connection.

For example, if the NST Linux IP address is 192.168.0.105 and the client is 192.168.0.104, run the *xhost* command from the customer side to include the NST Linux server in the rundown of permitted hosts to make the connections.

```
$ xhost +192.168.0.105
```

Once you have done this, you can attempt to dispatch the X Window application from the NST WUI. For instance, we can attempt dispatching the Wireshark application from the NST WUI. Go to the menu and then perform the following steps: *X > Network Applications > Wireshark (Packet Capture).*

At this point, Wireshark will show up.

The status (on probe) at the header reveals to us that it really begins from the server, yet is rendered on the client side. On the off chance that you are running the Microsoft Windows client, you can do the same on it, if you also run Cygwin/X on your Windows client. Reboot or shut down the server.

NST WUI also allows the server administrator to restart or shut down the server from the Web. If a server reboot is required, go to *System > Control Management > Reboot.*

Using NST in the wild

Let's now look at the different uses of NST in a wide variety of network environments.

Basic use case 1: This is a simple configuration for NST. A small computer like a notebook is attached directly to a broadband cable network. This configuration is helpful for checking and exploring the Intrusion Detection System (IDS).

Basic use case 2: There is another basic simple configuration, which involves a notebook computer running NST behind a router, switch, firewall or wireless device that is attached to a broadband cable network. This set-up is valuable for investigating the NST Linux operating system and its abilities.

Mobile wireless monitoring: This involves a notebook computer running NST to monitor 802.11 wireless networks. This plan is alluring for running the Kismet 3 remote network sniffers.

Small business configuration: This is a commonplace NST set-up and provides network security observation inside a private business network environment that is joined to the public Internet.

Enterprise configuration: This type of configuration gathers all the information from the network. A corporate enterprise network helps connect computers and related devices across departments and workgroups. For securing data or information, NST can be integrated in the corporate enterprise network environment.

Using VPNs with NST

VPN tunnels have for some time been used to secure and ensure the integrity of information over untrusted systems like the Internet. By using distinctive VPN connection types with NST, what you access on the Web and anything sent from one system to another is encrypted and directed through the VPN. So, data sent to the network can't be read by anyone except the VPN provider. When a connection is not encrypted, an attacker could perform a man-in-the-middle exploit (MITM), whereby the attacker can see all the information that is not encrypted being sent to other networks—including user names and passwords.

Let's now look at an instance of using VPN with Point to Point Protocol (PPP) over SSH.

VPN: PPP tunnelled over SSH

Probably the most common VPN solutions are SSH-Tunnel, PPP, PPTP and OpenVPN. I personally think that OpenVPN is the best option since it's strong and secure. The Secure Shell (SSH) can likewise be used to make an encoded tunnel between two PCs. PPP over SSH is a fast and speedy VPN solution. You can run a PPP connection over an SSH connection to make a simple, encrypted VPN. With PPP and SSH tools, you can create your own VPN within just a few minutes. Before I get into the details, shown below is a sample of PPP over SSH in a network.

Here, I will use a Windows machine to demonstrate the PPP tunnel over SSH in a VPN network. The steps are as follows.

1. Set up the VPN. The commands given below should run on a remote NST probe.

```
Script:"vpn-pppss" Is Run On Remote NST probe:"192.168.1.51"
SSH Configuration File:/root/.ssh/config
HOST nstprobe
HostName=70.22.33.10
Port=20022
VPN PPP SSH Script: vpn-pppssh
Vpn-ppssh -r nstprobe -s 172.18.2.31 -c 172.18.2.32 -rt -sn \
172.18.2.0/24 -cn 192.168.1.0/24 -nt -v
```

2. Once the VPN is set up, you can connect the two NST

probes between the two sites by using the point-to-point layered protocol.

3. As a tunnelling step CIFS share is mapped securely over the Internet. This tunnelling demonstrates the use of extending Corporate CIFS (SMB) file services to a remote satellite office securely over the untrusted public Internet.

Virtual computing

Virtual computing allows PC users remote access to programming applications and procedures when they require it. Users gain access via the Internet through a remote or network server.

Secure virtual computing

Secure virtual computing can be effected by tunnelling services or application protocols within an encrypted secure shell (SSH) session envelope.

Secure virtual computing with Microsoft Remote Desktop Protocol (RDP)

Two NST probes can be arranged for a VPN that tunnels the Remote Desktop Protocol between a terminal services server and a Microsoft terminal service client (mstsc) across the public Internet.

Note: The ISO image of NST 24 can be found with the bundled DVD.

END

By: Prerna Maheshwari

The author has a masters in commerce and is a technology enthusiast. Writing is her hobby. She can be contacted at *ca.prernamaheshwari@gmail.com.*

Continued from page 39...

You can now issue the following command to get the same result:

```
$ ansible ubuntu -m ping -u xetex

ubuntu | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

For the Ubuntu system, you can also add the defined user in the */etc/ansible/hosts* file as follows:

```
ubuntu ansible_ssh_host=ubuntu ansible_ssh_user=xetex
```

The ping command is now simplified to:

```
$  ansible ubuntu -m ping

ubuntu | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

You can now try the earlier Ansible commands on the target Ubuntu VM as illustrated below:

```
$ ansible ubuntu -a uptime

ubuntu | SUCCESS | rc=0 >>
 12:32:14 up 25 min,  3 users,  load average: 0.02, 0.07, 
0.06
```

```
$ ansible ubuntu -a date

ubuntu | SUCCESS | rc=0 >>
Sun Feb  5 12:32:45 IST 2017
$  ansible ubuntu -m setup
ubuntu | SUCCESS => {
    "ansible_facts": {
        "ansible_all_ipv4_addresses": [
            "192.168.122.250"
        ],
        "ansible_all_ipv6_addresses": [
            "ff20::5034:ff:fa9f:6123"
        ],
        "ansible_architecture": "x86_64",
        "ansible_bios_date": "04/01/2014",
        "ansible_bios_version": 
"1.10.1-20151022_124906-anatol",
        "ansible_cmdline": {
            "BOOT_IMAGE": "/boot/vmlinuz-3.19.0-15-generic",
            "quiet": true,
            "ro": true,
            "root": "UUID=f43c2c72-5bc7-4a97-9a43-
12e634ae232af",
            "splash": true,
            "vt.handoff": "7"
        },
        …
```

END

By: Shakthi Kannan

The author is a free software enthusiast and blogs at *shakthimaan.com.*

A Beginner's Guide to Mininet

You can instantly create realistic virtual networks deploying controllers, switches and hosts using Mininet. And experiment with them to your heart's content to run real kernel, application and switch code on a single machine, whether on a VM, the cloud or native.

Mininet is open source software that is used to simulate a software defined network (SDN) and its compatible controllers, switches and hosts. It comes in handy for networking enthusiasts and beginners to get some experience in working with the fast growing SDN technology. To put it simply, a SDN separates the control plane from the data forwarding plane, making network devices such as switches and routers fully programmable and, hence, the network behaves according to the users' requirements. By default, Mininet provides OVS switches and OVS controllers. However, it has the support to install other/preferred SDN controllers and switches instead of the defaults. The primary feature that distinguishes SDN devices from traditional network devices is the scope for customising protocols and functions.

Mininet supports the Openflow protocol, which provides an interface between the control plane and the data forwarding plane. Openflow protocols are used to control the packet flow as per the API written on the controller. Mininet also has support for a variety of topologies and ensures the availability of custom topologies. The CLI (command line interface) provided by Mininet is comfortable to use after a bit of practice.

Installing Mininet on your computer

To install Mininet, open the terminal and issue the following command:

```
# apt-get install mininet , to install the Mininet packages
on your system .
```

Next, verify the installation by issuing the following command:

```
 # mn
```

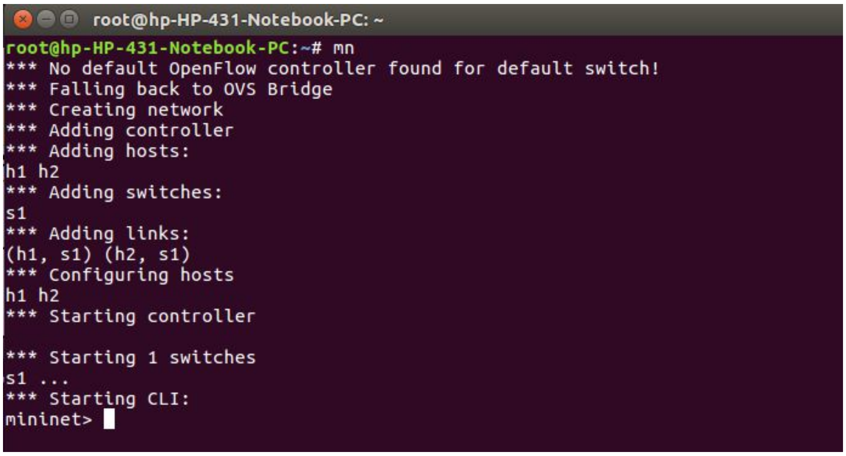
After a successful installation, you'll see what's shown in Figure 1 on the screen.

Notice that a network is created with default topology (as shown in Figure 2) and default OVS switches. This network is ready for use and has all the parameters like IP addresses and links pre-configured, based on the default settings.

Getting started with the Mininet commands

The command to display the nodes present in the network is:

```
mininet> nodes
```



```
root@hp-HP-431-Notebook-PC: ~
root@hp-HP-431-Notebook-PC:~# mn
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> 
```

Figure 1: Mininet in action

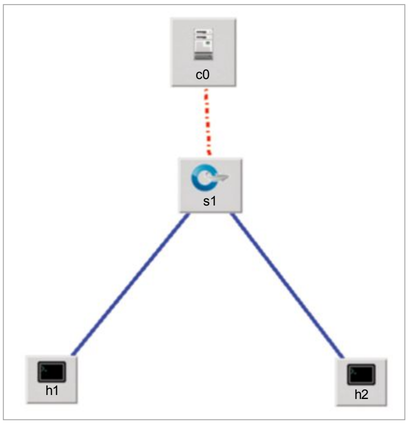
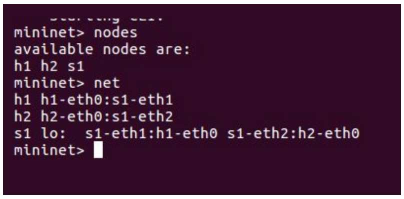
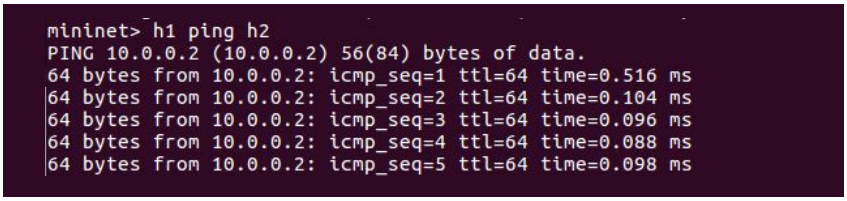


Figure 2: Topology created using Miniedit (an open source tool to create Mininet topologies)



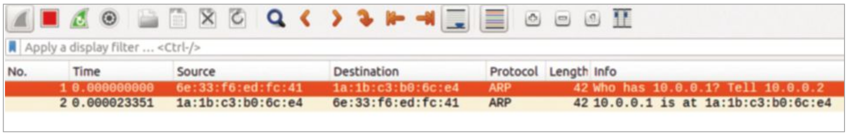
```
mininet> nodes
available nodes are:
h1 h2 s1
mininet> net
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
s1 lo:  s1-eth1:h1-eth0 s1-eth2:h2-eth0
mininet> 
```

Figure 3: Nodes and net



```
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.516 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.104 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.096 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.088 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=0.098 ms
```

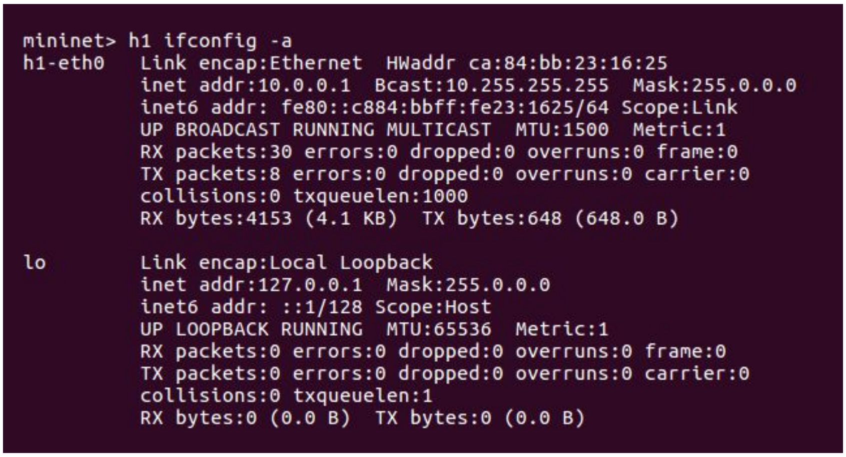
Figure 4: The command makes *h1* ping *h2*



Apply a display filter ... <Ctrl-/>

```
No.  Time         Source             Destination        Protocol Length Info
   1 0.000000000  6e:33:f6:ed:fc:41  1a:1b:c3:b0:6c:e4  ARP          42 Who has 10.0.0.1? Tell 10.0.0.2
   2 0.000023351  1a:1b:c3:b0:6c:e4  6e:33:f6:ed:fc:41  ARP          42 10.0.0.1 is at 1a:1b:c3:b0:6c:e4
```

Figure 5: Packet captured at the interface *s1-eth1.*



```
mininet> h1 ifconfig -a
h1-eth0   Link encap:Ethernet  HWaddr ca:84:bb:23:16:25
          inet addr:10.0.0.1  Bcast:10.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::c884:bbff:fe23:1625/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:30 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4153 (4.1 KB)  TX bytes:648 (648.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Figure 6: Getting details of host1 interface

The above command will list all the nodes present in the created network. As shown in Figure 3, the nodes s1, h1, h2 are displayed.

The command to display and list the links present in the network is:

```
Mininet>net
```

As shown in Figure 3, the interface *eth0* of the host *h1* is connected to *eth1* of switch *s1* and the interface *eth0* of host *h2* is connected to *eth2* of switch *s2*.

The command to display the IP addresses and the process IDs of the nodes is:

```
Mininet>dump
```

As seen in Figure 3, *h1* is assigned the IP address 10.0.0.1 with the process ID 4403 and *h2* is assigned the IP address 10.0.0.2 with the process ID 4407.

- The command to ping a specific host to a targeted host is:

```
Mininet> h1 ping h2
```

On inspecting the packets, we can see that the first ping has taken considerably longer (0.805) than the others. This is because ARP tables, MAC tables, etc, are initialised during the first ping.

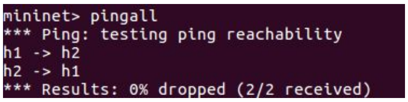
The command to display the address information of the nodes is:

```
Mininet> h1 ifconfig -a
```

This command will display the IP address, broadcast address and MAC address of the host *h1*, as in Figure 6.

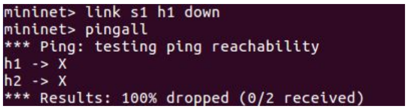
- The command to test the connectivity among hosts is:

```
Mininet>pingall
```



```
mininet> pingall
*** Ping: testing ping reachability
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
```

Figure 7: Ping traceability for host1 and host2



```
mininet> link s1 h1 down
mininet> pingall
*** Ping: testing ping reachability
h1 -> X
h2 -> X
*** Results: 100% dropped (0/2 received)
```

Figure 9: Bringing down the interface for host 1 connected to switch 1

This command will make each host in the network ping every other host in the network. In the network that we have, *h1* will ping *h2*, and *h2* will ping *h1*. As seen in Figure 7, the successful pings indicate that all the links in the network are active.

When we investigate the interface *eth1* of switch *s1* using Wireshark, we find that both the ping requests are successful (Figure 8).

The command to down a link is:

```
Mininet> link s1 h1 down
```

The above command will down the link between switch *s1* and host *h1* (Figure 9). Further, on pinging the hosts using the *pingall* command, we can see that both the pings are unsuccessful due to the link getting down.

- The command to build custom topology is:

```
#sudo mn –topo single,3
```

This command will create a topology as shown in Figure 12, and initialise the links and addresses of hosts and switches.

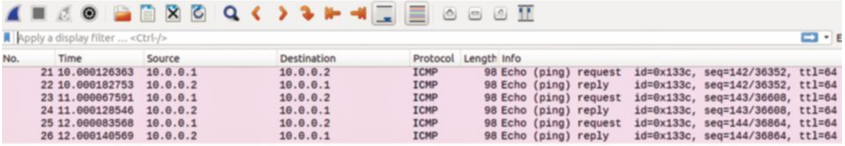
- The command to perform regression testing is:

```
#sudo mn -- test pingpair
```

The above command is used to create a network with default topology, perform the *pingall* function and stop the network. This command is basically used to test how Mininet works.

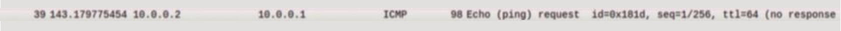
- The command to open the xterm window is:

```
Mininet> h1 xterm
```



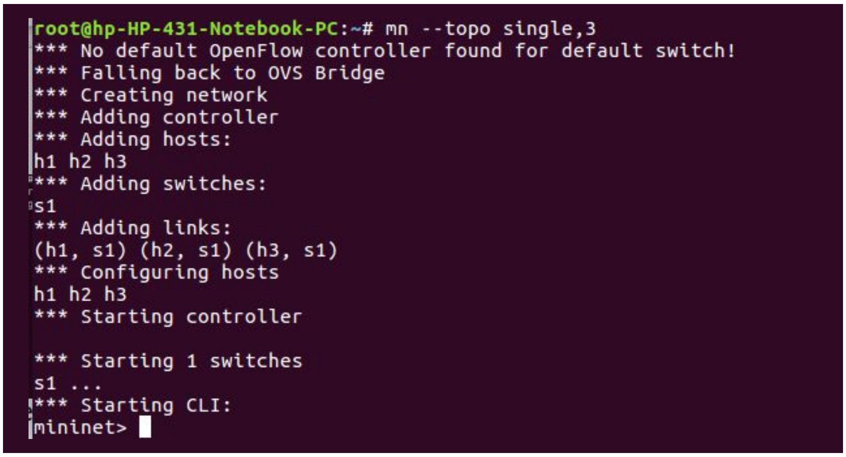
```
No.   Time            Source      Destination   Protocol  Length  Info
   21 10.000126363    10.0.0.1    10.0.0.2      ICMP      98 Echo (ping) request  id=0x133c, seq=142/36352, ttl=64
   22 10.000182753    10.0.0.2    10.0.0.1      ICMP      98 Echo (ping) reply    id=0x133c, seq=142/36352, ttl=64
   23 11.000067591    10.0.0.1    10.0.0.2      ICMP      98 Echo (ping) request  id=0x133c, seq=143/36608, ttl=64
   24 11.000128546    10.0.0.2    10.0.0.1      ICMP      98 Echo (ping) reply    id=0x133c, seq=143/36608, ttl=64
   25 12.000083568    10.0.0.1    10.0.0.2      ICMP      98 Echo (ping) request  id=0x133c, seq=144/36864, ttl=64
   26 12.000140569    10.0.0.2    10.0.0.1      ICMP      98 Echo (ping) reply    id=0x133c, seq=144/36864, ttl=64
```

Figure 8: Ping packets (ICMP) captured



```
   39 143.179775454 10.0.0.2        10.0.0.1        ICMP      98 Echo (ping) request  id=0x181d, seq=1/256, ttl=64 (no response
```

Figure10: Unsuccessful ping packets captured at interface *s1-eth1*



```
root@hp-HP-431-Notebook-PC:~# mn --topo single,3
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller

*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> 
```

Figure 11: Topology created in Mininet

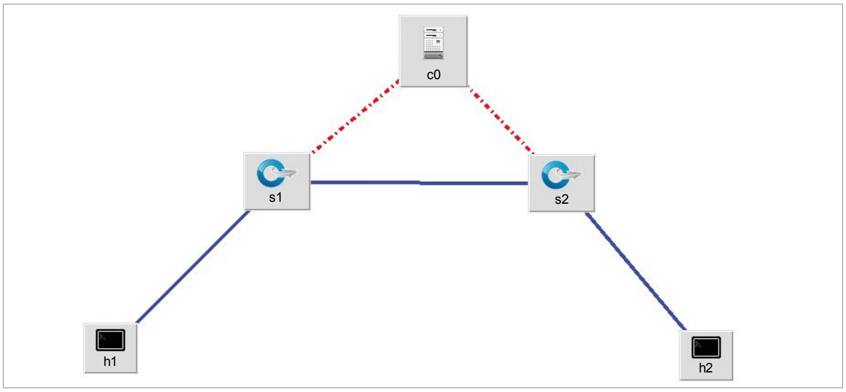
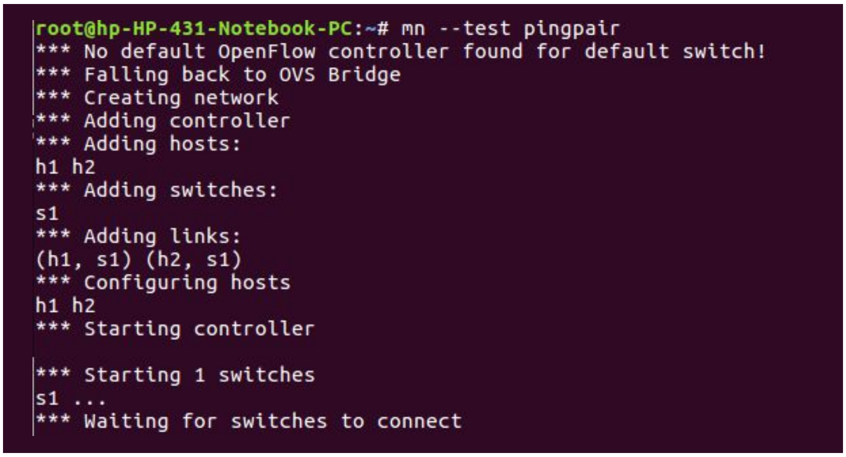


Figure 12: Topology creation using Miniedit (an open source tool to create Mininet topologies)



```
root@hp-HP-431-Notebook-PC:~# mn --test pingpair
*** No default OpenFlow controller found for default switch!
*** Falling back to OVS Bridge
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1)
*** Configuring hosts
h1 h2
*** Starting controller

*** Starting 1 switches
s1 ...
*** Waiting for switches to connect
```

Figure 13a: Topology creation and validation tests at mining

Gift Yourself Knowledge

And get more gifts from us

OFFERS FOR YOU & ME

Portronics

YOGG Wrist Band

Your personal fitness and activity tracker

10W LED Bulb
Branded 10W LED bulb with a two-year warranty

USB Car Charger
Three USB ports with micro USB cable

10W LED Bulb + Three Night Lamps
Set of one 10W LED bulb with three 0.5W night lamps

ORDER FORM

Magazine	Duration	Cover Price (₹)	You Pay (₹)	You Save (₹)	Gifts For You & Me
Electronics For You	1 Year	720	625	95	□ 10W LED Bulb
	2 Years	1440	1150	290	□ USB Car Charger
	3 Years	2160	1725	435	□ 10W LED Bulb+3 Night Lamps
	5 Years	3600	2880	720	□ Yogg Smart Wristband

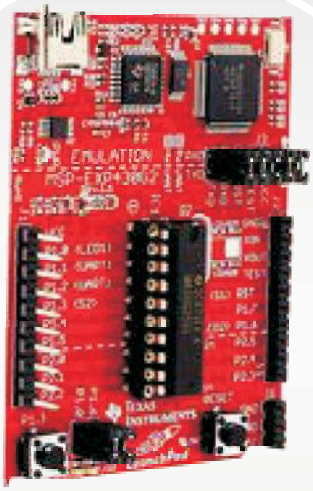
To Subscribe Online, visit:
http://subscribe.efyindia.com

Technology Advisers for Businesses and Techies

₹ 60
OCTOBER 2015
electronics FOR YOU
SOUTH ASIA'S MOST POPULAR ELECTRONICS MAGAZINE
Are Robots Replacing Humans?
TOP 5 DO-IT-YOURSELF

SUBSCRIBE TO ELECTRONICS FOR YOU,
SAVE MONEY & GET GIFTS

OFFERS FOR TECHIES

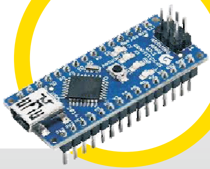


Texas Instrument's MSP430 Launchpad

At the heart of the Texas Instruments' LaunchPad range of development boards is the MSP430. The MSP430 is an easy-to-use flash programming microcontroller, just like other microcontrollers such as the ATmega range from Atmel. It was specifically designed to be low-cost and low-powered, suitable for a range of applications.

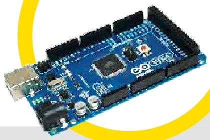
Arduino Nano

Arduino Nano is a compact and breadboard-friendly version board based on ATmega328 processor.



Arduino Mega

The MEGA is designed for more complex projects with 54 digital I/O pins, 16 analog inputs and a larger space for your sketch it is the recommended board for 3D printers and robotics projects.



Arduino Experiment Kit

An awesome Arduino Experiment kit with which you can learn and develop projects in a very efficient manner.

ORDER FORM

Magazine	Duration	Cover Price (₹)	You Pay (₹)	Gifts For Techies
Electronics For You	1 Year	720	720	□ Arduino Nano
	2 Years	1440	1440	□ Arduino Mega
	3 Years	2880	2500	□ Texas Instrument's MSP430 Launchpad
	5 Years	3600	3200	□ Arduino Experiment Kit

Free ezine
Access With
Every Subscription

Name____________________Designation____________________Organisation____________________

Mailing Address____

City____________Pin Code________ State__________Phone/Mobile______________Email________________

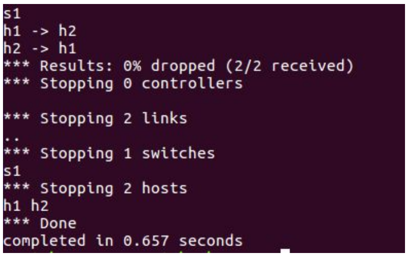
Subscription No. (for existing subscribers only)______________. **I would like to subscribe to Electronics For You starting with the next issue.** Please find enclosed a sum of

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)*

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: **011-26810601-03** | Fax: **011-26817563** | e-mail: **info@efy.in** | **www.efy.in**

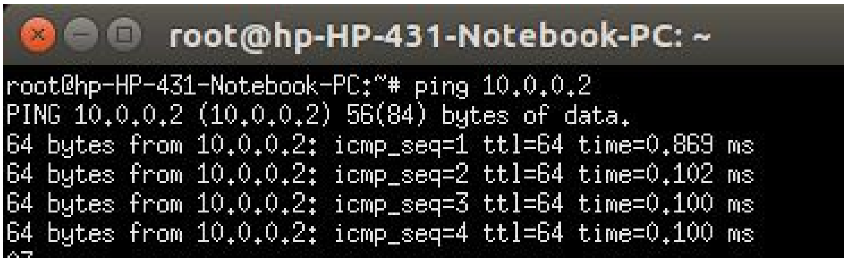
EFYGROUP
Technology Drives Us

Terms & conditions: # This offer is applicable for new subscriptions and renewal by existing subscribers. # The rates are valid for subscribers within India only. # Please allow 4-6 weeks for processing your subscription and 2 weeks to send your gifts # Replacement copies will be sent only if intimation of damaged / non-receipt of copies is received within 15 days of publication # If you are not satisfied with the magazine or our services, we shall refund the balance amount after three months. # Disputes, if any, are subject to exclusive jurisdiction of competent courts and forums in Delhi/New Delhi only.



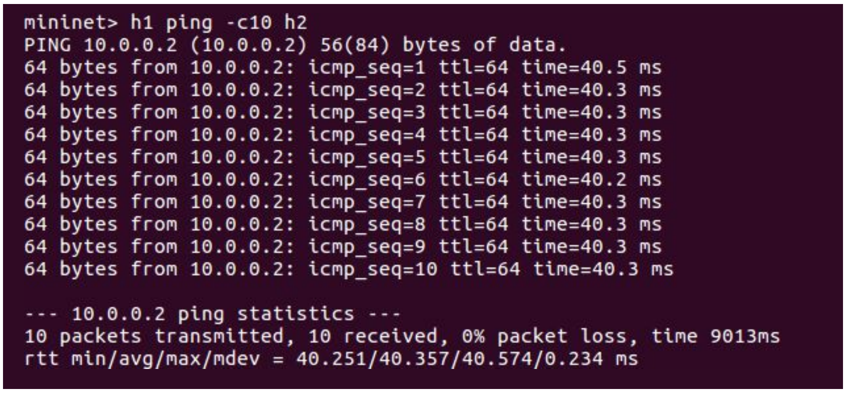
```
s1
h1 -> h2
h2 -> h1
*** Results: 0% dropped (2/2 received)
*** Stopping 0 controllers

*** Stopping 2 links
..
*** Stopping 1 switches
s1
*** Stopping 2 hosts
h1 h2
*** Done
completed in 0.657 seconds
```

Figure 13b: Topology creation and validation tests at mining



```
root@hp-HP-431-Notebook-PC:~# ping 10.0.0.2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=0.869 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.102 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=0.100 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=0.100 ms
```

Figure 14: Testing the availability of host *h2* from host *h1*



```
mininet> h1 ping -c10 h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=40.5 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=40.3 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=64 time=40.3 ms
64 bytes from 10.0.0.2: icmp_seq=4 ttl=64 time=40.3 ms
64 bytes from 10.0.0.2: icmp_seq=5 ttl=64 time=40.3 ms
64 bytes from 10.0.0.2: icmp_seq=6 ttl=64 time=40.2 ms
64 bytes from 10.0.0.2: icmp_seq=7 ttl=64 time=40.3 ms
64 bytes from 10.0.0.2: icmp_seq=8 ttl=64 time=40.3 ms
64 bytes from 10.0.0.2: icmp_seq=9 ttl=64 time=40.3 ms
64 bytes from 10.0.0.2: icmp_seq=10 ttl=64 time=40.3 ms

--- 10.0.0.2 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9013ms
rtt min/avg/max/mdev = 40.251/40.357/40.574/0.234 ms
```

Figure 15: Delay induced ping

xterm is the command used to initiate an external and separate terminal. The command opens an xterm window, which is specific to a node in the network. Several functionalities specific to a node can be implemented on this window. In the example shown in Figure 14, the *h1* xterm window is used to ping *h2* (10.0.0.2).

- The command to provide a delay of 10ms and set a bandwidth of 10Mbps in all the links is:

```
#sudo mn -link tc,bw=10,delay=10ms
```

Notice that *h1* pinging *h2* takes a total time of 40ms (approximately), since it has to travel through the *s1-h1 link*. END

By: Raghul S., T. Subashri and K.R. Vimal

Raghul S. is a technology enthusiast. He can be reached at *raghulsekar77(AT)gmail(DOT)com.*

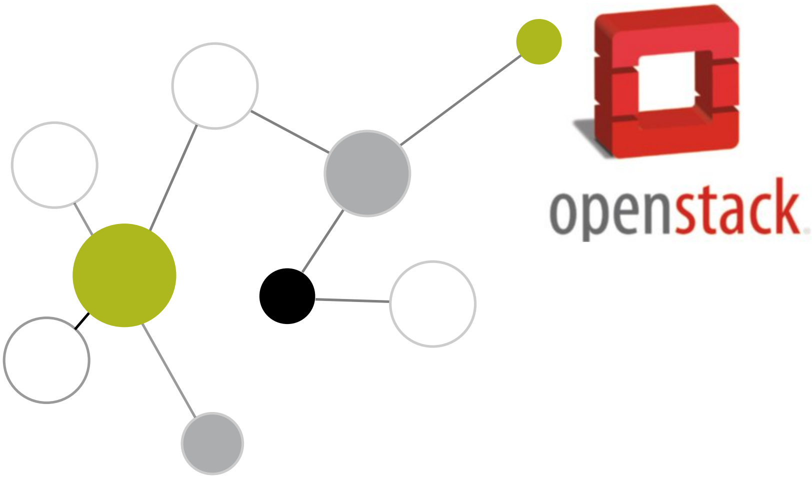
T. Subashri works as an assistant professor in the department of electronics engineering, MIT, Anna University, Chennai. She can be contacted at *tsubashri(AT)annauniv(DOT)edu.*

K.R. Vimal works as a systems admin for the high performance computing research lab at MIT, Anna University, Chennai. He is also an authorised CCNA course instructor. He can be reached at *vimalkrme(AT)gmail(DOT)com.*

OSFY Magazine Attractions During 2017-18

MONTH	THEME
March 2017	Open Source Firewall, Network security and Monitoring
April 2017	Databases management and Optimisation
May 2017	Open Source Programming (Languages and tools)
June 2017	Open Source and IoT
July 2017	Mobile App Development and Optimisation
August 2017	Docker and Containers
September 2017	Web and desktop app Development
October 2017	Artificial Intelligence, Deep learning and Machine Learning
November 2017	Open Source on Windows
December 2017	BigData, Hadoop, PaaS, SaaS, Iaas and Cloud
January 2018	Data Security, Storage and Backup
February 2018	Best in the world of Open Source (Tools and Services)

Integrating OpenDaylight VTN Manager with OpenStack



OpenDaylight is the largest open source SDN controller. The OpenDaylight virtual tenant network (VTN) is an application that provides a multi-tenant virtual network on an SDN controller. This article is a tutorial on how to integrate the OpenDaylight VTN Manager with OpenStack.

A virtual tenant network (VTN) allows users to define the network with a look and feel of the conventional L2/L3 network. Once the network is designed on VTN, it automatically gets mapped into the underlying physical network, and is then configured on the individual switch, leveraging the SDN control protocol. The definition of the logical plane makes it possible not only to hide the complexity of the underlying network but also to manage network resources better. This reduces the reconfiguration time of network services and minimises network configuration errors.

The technical introduction given above might seem complex to SDN beginners. In this article, I have tried to be as simple as I can, while teaching readers about VTN and its integration with OpenStack.

For the purpose of this article, I'm assuming readers have a basic understanding of SDN. So let me start with VTN directly.

The SDN VTN Manager helps you to aggregate multiple ports from the many underlying SDN-managed switches (both physical and virtual) to form the single isolated virtual tenant network (VTN). Each tenant network has the capability to function as an individual switch.

For example, consider that you have two physical switches (say, s1 and s2) and one virtual Open vSwitch (say, vs1) in your lab environment. Now, with the help of the VTN Manager, it is possible to group (aggregate) the three ports (say p1, p2 and p3) from switch s1 — i.e., s1p1, s1p2, s1p3; two ports from switch s2 — i.e., s2p1 and s2p2; and two ports from the virtual switch vs1 — i.e., vs1p1 and vs2p2, to form a single switch environment (say, VTN-01).

This means, virtually, the group (tenant) named VTN-01 is one switch with seven ports (s1p1, s1p2, s1p3, s2p1, s2p2, vs1p1 and vs2p2) in it. This VTN-01 will act exactly like a single isolated switch with the help of flows configured in the ports of all three switches by the OpenDaylight VTN Manager.

The above example explains the concept called port mapping in VTN, and will help beginners to understand the basic concept better. It will also help them to compare all other VTN concepts like VLAN mapping and MAC mapping.

VTN OpenStack integration

There are several ways to integrate OpenDaylight with OpenStack. This article will focus on the method that uses VTN features available on the OpenDaylight controller. During integration, the VTN Manager works as the network service provider for OpenStack.

The features of VTN Manager empower OpenStack to work in a pure OpenFlow environment, in which all the switches in the data plane are an OpenFlow switches. You could also refer to my blog on 'OpenDaylight Integration with OpenStack using OVSDB' from the link

http://www.cloudenablers.com/blog/opendaylight-integration-with-openstack/.

The requirements are:

- OpenDaylight Controller
- OpenStack Control Node
- OpenStack Compute Node

OpenDaylight support for OpenStack network types

Till the Boron release, OpenDaylight (ODL) only supported‘Local’network type in OpenStack and there was no support for VLAN. You may wonder why the developers never speak about VxLAN and GRE tunnelling network types support. You can answer that question if you recall the example I mentioned at the beginning of this article.

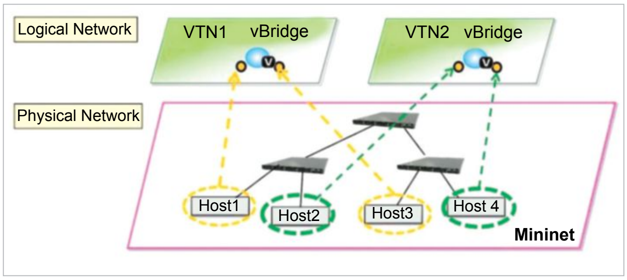
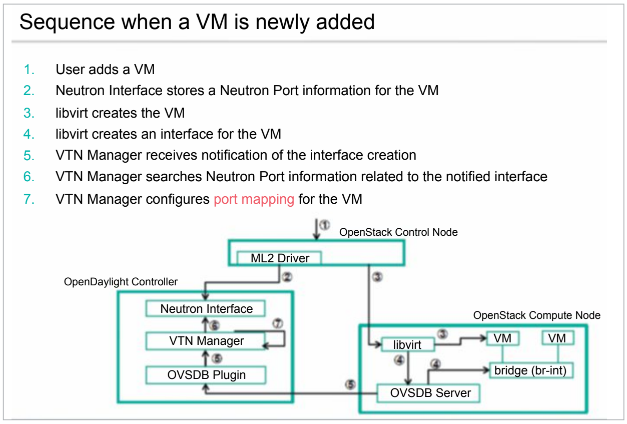


Figure 1: Virtual Tenant Network


Sequence when a VM is newly added

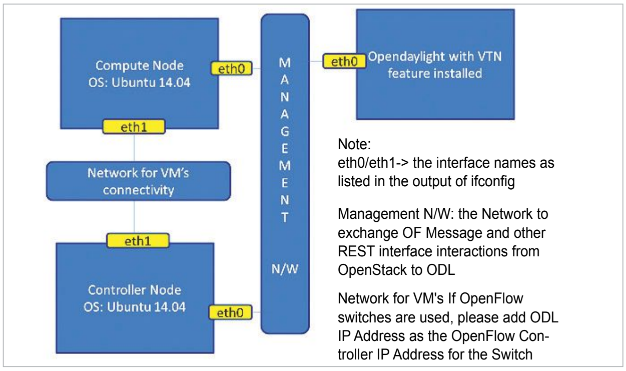
1. User adds a VM
2. Neutron Interface stores a Neutron Port information for the VM
3. libvirt creates the VM
4. libvirt creates an interface for the VM
5. VTN Manager receives notification of the interface creation
6. VTN Manager searches Neutron Port information related to the notified interface
7. VTN Manager configures port mapping for the VM

Figure 2: Request flow

To recap, I said that with the help of the VTN Manager, the user can group multiple ports from multiple switches in the infrastructure to form a single isolated network.

Let's compare this with our OpenStack environment, which has two Open vSwitches installed in the controller and compute node.

1. Whenever a new network is created in OpenStack, VTN Manager creates a new VTN in ODL.
2. Whenever a new sub-network is created, VTN Manager handles it and creates a vBridge under the VTN. vBridge is nothing but the virtual switch.
3. When a new VM is created in OpenStack, the addition of a new port in the Open vSwitch of the compute node is captured by VTN Manager, and it creates a vBridge interface in the newly created vBridge and maps that Open vSwitch port with the particular vBridge port.
4. In this case, the port (say, vs1p1) of the DHCP agent in the Open vSwitch of the controller node and the port (vs2p1) of the VM in the compute node are isolated from the actual Open vSwitch, using the flow entries from the OpenDaylight VTN Manager, to form a new virtual switch environment called the virtual tenant network.
5. When the packet sent from the DHCP agent reaches the OpenStack controller's Open vSwitch port vs1p1, then flow entries will tell the port vs1p1 to forward the packet to the compute node's Open vSwitch port vs2p1 using the underlying physical network. This packet will be sent as a regular TCP packet with a source and destination MAC address, which means that the traffic created in one network can be sent as a regular packet across the controller and compute node without any tunnelling protocol.
6. This explains why support for VxLAN and GRE network types is not required.



Note:
eth0/eth1-> the interface names as listed in the output of ifconfig

Management N/W: the Network to exchange OF Message and other REST interface interactions from OpenStack to ODL

Network for VM's If OpenFlow switches are used, please add ODL IP Address as the OpenFlow Controller IP Address for the Switch

Figure 3: LAB layout

LAB set-up layout

The VTN features support multiple OpenStack nodes. Hence, you can deploy multiple OpenStack compute nodes.

In the management plane, OpenDaylight controller, OpenStack nodes and OpenFlow switches (optional) should communicate with each other.

In the data plane, Open vSwitches running in OpenStack nodes should communicate with each other through physical or logical OpenFlow switches (optional). Core OpenFlow switches are not mandatory. Therefore, you can directly connect to the Open vSwitches.

You may need to disable the firewall (UFW) in all the nodes to reduce the complexity.

Installing OpenStack with the Open vSwitch configuration

Installing OpenStack is beyond the scope of this article; however, getting started with a minimal multi-node OpenStack deployment is recommended.

To help speed up the process, you could use my fully automated bash script for installing the OpenStack-Mitaka set-up at *https://github.com/CloudenablersPvtLtd/openstack-setup.*

Note: This script will install OpenStack and configure the Linux bridge for networking. But for the VTN integration to work in OpenStack, we need network configuration with Open vSwitch. So, you must uninstall the Linux bridge settings and reconfigure with Open vSwitch.

After the successful OpenStack installation, run the sanity test by performing the following operations.

Create two instances on a private subnet. Then add the floating IP address from your public network, verify that you can connect to them and that they can ping each other.

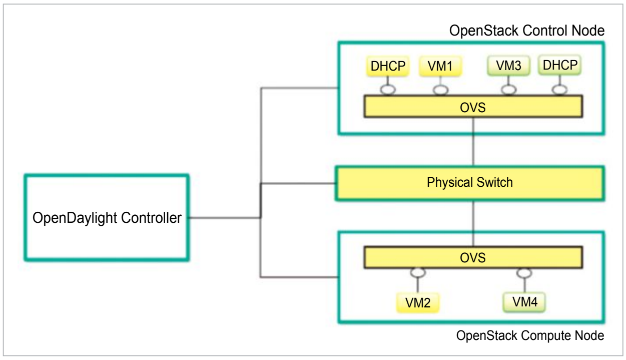


Figure 4: VTN OpenStack architecture

Installing OpenDaylight

The OpenDaylight controller runs in a JVM. The OpenDaylight-Boron release requires OpenJDK8, which you can install using the command given below:

```
$apt-get install openjdk-8-jdk
```

Download the latest OpenDaylight-Boron package from the official repo, as follows:

```
$wget https://nexus.opendaylight.org/content/repositories/
opendaylight.release/org/opendaylight/integration/
distribution-karaf/0.5.1-Boron-SR1/distribution-karaf-0.5.1-
Boron-SR1.tar.gz
```

Untar the file as the root user, and start OpenDaylight using the commands given below:

```
$ tar -xvf distribution-karaf-0.5.1-Boron.tar.gz
$ cd distribution-karaf-0.5.1-Boron.tar.gz
$ ./bin/karaf
```

Now, you should be in OpenDaylight's console. Install all the required features, as follows:

```
opendaylight-user@root> feature:install odl-vtn-manager-
neutron
opendaylight-user@root> feature:install odl-vtn-manager-rest
opendaylight-user@root> feature:install odl-mdsal-apidocs
opendaylight-user@root> feature:install odl-dlux-all
```

Feature installation may take some time. Once the installation is complete, you can check whether everything is working fine by using the following *curl* call:

```
$ curl -u admin:admin http://<ODL_IP>:8080/controller/nb/v2/
neutron/networks
```

The response should be an empty network list if OpenDaylight is working properly.

Now, you should be able to log into the DLUX interface on *http://<ODL_IP>:8181/index.html.*

The default username and password are *admin/ admin*.

Additionally, you could find useful log details at the following location:

```
$ tail -f /<directory_of_odl>/data/log/karaf.log
$ tail -f /<directory_of_odl>/logs/web_access_log_2015-12.txt
```

Now, you have a working OpenDaylight-Boron set-up. Let's get into the integration part.

Configuring OpenStack for VTN integration

Step 1

Erase all VMs, networks, routers and ports in the controller node, since you already have a working OpenStack set-up. You might test for VM provisioning as a sanity test, but before integrating OpenStack with OpenDaylight, you must clean up all the unwanted data from the OpenStack database. When using OpenDaylight as the Neutron back-end, ODL expects to be the only source for Open vSwitch configuration. Because of this, it is necessary to remove existing OpenStack and Open vSwitch settings to give OpenDaylight a clean slate.

The following steps will guide you through the cleaning process.

- Delete instances, as follows:

```
$ nova list
$ nova delete <instance names>
```

- Remove links from subnets to routers, as follows:

```
$ neutron subnet-list
$ neutron router-list
$ neutron router-port-list <router name>
$ neutron router-interface-delete <router name> <subnet ID or name>
```

- Delete subnets, nets and routers, as follows:

```
$ neutron subnet-delete <subnet name>
$ neutron net-list
$ neutron net-delete <net name>
$ neutron router-delete <router name>
```

- Check that all ports have been cleared – at this point, this should be an empty list:

```
$ neutron port-list
```

- Stop the Neutron service, as follows:

```
$ service neutron-server stop
```

While Neutron is managing the OVS instances on the compute and control nodes, OpenDaylight and Neutron may be in conflict. To prevent issues, let's turn off the Neutron server on the network controller and Neutron's Open vSwitch agents on all hosts.

Step 2: Configuring Open vSwitches in the controller and compute nodes

The Neutron plugin in every node must be removed because only OpenDaylight will be controlling the Open vSwitches. So, on each host, we will erase the pre-existing Open vSwitch config and set OpenDaylight to manage the Open vSwitch:

```
$ apt-get purge neutron-plugin-openvswitch-agent
$ service openvswitch-switch stop
$ rm -rf /var/log/openvswitch/*
$ rm -rf /etc/openvswitch/conf.db
$ service openvswitch-switch start
$ ovs-vsctl show
# The above command must return the empty set except
OpenVswitch ID and it's Version.
```

Step 3:Connecting Open vSwitch to OpenDaylight

Use the command given below to make OpenDaylight administer Open vSwitch:

```
$ ovs-vsctl set-manager tcp:<OPENDAYLIGHT MANAGEMENT IP>:6640
```

You can copy the Open vSwitch ID from the command *ovs-vsctl show*. Execute the above command in all the nodes (controller and compute nodes) to set ODL as the manager for Open vSwitch:

```
$ ovs-vsctl show
```

The above command will show that you are connected to the OpenDaylight server, which will automatically create a *br-int* bridge.

```
[root@vinoth ~]# ovs-vsctl show
9e3b34cb-fefc-4br4-828s-084b3e55rtfd
Manager "tcp:192.168.2.101:6640"
Is_connected: true
Bridge br-int
Controller "tcp:192.168.2.101:6633"
fail_mode: secure
Port br-int
Interface br-int
ovs_version: "2.1.3"
```

If you get any error messages during bridge creation, you may need to log out from the OpenDaylight Karaf console and check the *90-vtn-neutron.xml* file from the following path *distribution-karaf-0.5.0-Boron/etc/opendaylight/karaf/*.

The contents of *90-vtn-neutron.xml* should be as follows:

```
bridgename=br-int
portname=eth1
protocols=OpenFlow13
failmode=secure
```

By default, if *90-vtn-neutron.xml* is not created, VTN uses *ens33* as the port name.

After running the ODL controller, please ensure it listens to the following ports: 6633, 6653, 6640 and 8080.

Note:

- 6633/6653 are the OpenFlow ports.
- 6640 is the OVS Manager port.
- 8080 is the port for the REST API.

Step 4: Configure ml2_conf.ini *for the ODL driver*

Edit *vi /etc/neutron/plugins/ml2/ml2_conf.ini* in all the required nodes and modify the following configuration. Leave the other configurations as they are.

```
[ml2]
type_drivers = local
tenant_network_types = local
mechanism_drivers = opendaylight
[ml2_odl]
password = admin
username = admin
url = http://<OPENDAYLIGHT SERVER's IP>:8080/controller/nb/
v2/neutron
```

Step 5: Configure the Neutron database

Reset the Neutron database, as follows:

```
$ mysql -uroot -p
$ drop database neutron;
$ create database neutron;
$ grant all privileges on neutron.* to 'neutron'@'localhost'
identified by '<YOUR NEUTRON PASSWORD>';
$ grant all privileges on neutron.* to 'neutron'@'%'
identified by '<YOUR NEUTRON PASSWORD>';
$ exit
$ su -s /bin/sh -c "neutron-db-manage --config-file /etc/
neutron/neutron.conf --config-file /etc/neutron/plugins/ml2/
ml2_conf.ini upgrade head" neutron
```

Restart the Neutron-server, as follows:

```
$ service neutron-server start
```

Step 6: Install the Python-networking-odl *Python module*

IMPORTANT: You should get the status alert if the Neutron service fails to start by this time. Don't worry. This is a temporary issue since you have enabled OpenDaylight as the *mechanism_driver* but not yet installed the Python module for it.

Install the *Python-networking-odl* Python module, as follows:

```
$ apt-get install python-networking-odl
```

Now, restart the Neutron server and check its status. It should be running without errors.

Step 7: Verify the integration

We have almost completed the integration of OpenStack with VTN. Now, create initial networks in OpenStack and check whether a new network is created and posted to ODL, for which VTN Manager creates a VTN.

Use the *curl* commands given below to verify the creation of the network and VTN:

```
$ curl --user "admin":"admin" -H "Content-type: application/
json" -X GET http://<ODL_IP>:8181/restconf/operational/
vtn:vtns/
$ curl -u admin:admin http://<ODL_IP>:8080/controller/nb/v2/
neutron/ networks
```

Whenever a new sub-network is created in the OpenStack Horizon, VTN Manager will handle it and create a vBridge under the VTN. When you create a new VM in OpenStack, the interface (br-int) mentioned as the integration bridge in the configuration file will be added with more interfaces, and the network is provisioned for it by the VTN Neutron bundle. The addition of the new port is captured by VTN Manager, and it creates a vBridge interface with port mapping.

When the VM starts to communicate with the other VMs that have been created, VTN Manager will install flows in the OVS and other OpenFlow switches to facilitate communication between the VMs.

Note: To access OpenDaylight RestConf API documentation, use the link *http://<ODL_IP>:8181/apidoc/explorer/index.html*, which points to your *ODL_IP*.

If everything works correctly, you will able to communicate with other VMs created in the different compute nodes.

The VTN project doesn't support the vRouter up to the Boron release, which means that the floating IP operation in OpenStack is not supported when integrating VTN Manager with OpenStack. It might support the vRouter in the Carbon or Nitrogen releases. END

References

[1] *http://www.hellovinoth.com/*

[2] *http://www.cloudenablers.com/blog/*

[3] *https://www.opendaylight.org/*

[4] *http://docs.openstack.org/*

By: Vinoth Kumar Selvaraj

The author is a DevOps engineer at Cloudenablers Inc., a cloud technology startup based in Chennai. He has also worked as a book reviewer with PackPub Publishers, for books related to OpenStack. He blogs at *http://www.hellovinoth.com*. His Twitter handle is *@vinoth6664*.

Statement about ownership and other particulars about
OPEN SOURCE FOR YOU
FORM IV (See Rule 8)

1.	Place of publication	:	New Delhi
2.	Periodicity of its publication	:	Monthly
3.	Printer's Name	:	Ramesh Chopra
	Nationality	:	Indian
	Address	:	OPEN SOURCE FOR YOU D-87/1, Okhla Industrial Area, Phase-1, New Delhi 110020
4.	Publisher's Name Nationality and address	:	Same as (3) above
5.	Names and addresses of individuals who own the newspaper & partners or shareholders holding more than 1% of the total capital	:	**EFY Enterprises Pvt Ltd** D-87/1, Okhla Industrial Area, Phase-1, New Delhi 110020

I, Ramesh Chopra, hereby declare that the particulars given above are true to the best of my knowledge and belief.

Date: 28-2-2017

Ramesh Chopra
Publisher

An Overview of Open Source Tools for Network Monitoring

Network monitoring is crucial in enterprises. This article discusses its various advantages and highlights a few network monitoring tools. Systems admins will benefit greatly by familiarising themselves with these tools.

Today, computers, smartphones and the Internet have become our lifelines. Many of us consider them necessary for our day to day life, especially this revolutionary 4G phenomenon. We get annoyed the instant our mobile Internet network degrades from 4G to 3G, resulting in slow buffering when downloading any video. Hence, it has become important and also a bit challenging for different network providers to meet the expectations of their customers by continuously providing uninterrupted network services. In order to be true to customers' expectations, different network providers need to closely and continuously monitor the network services they provide to ensure network supply without any outages. Sometimes, an intermediate component may be slow or may even fail, leading to slow processing of data in the network. Since it is difficult to monitor and keep a track of each network service manually, different tools or systems are used to monitor performance and take corrective measures if it does not meet expectations.

According to Wikipedia, network monitoring is the usage of a system that constantly monitors a computer network for slow or failing components and notifies the network administrator via email, pager or other alarms in case of any outages or trouble.

In addition, network monitoring also takes care of the performance and utilisation of the network and predicts the possible outcomes of any threat to it, thereby preventing the system from a possible major outage in the future. Network failures, server downtime, service or application crashes can seriously threaten a business' operations, resulting in the loss of thousands of rupees in revenue and productivity. Hence, by using network monitoring solutions, a company can deliver a better service as well as cut costs by fixing issues before any of its users notice a problem.

Why network monitoring?

Since we're today well aware of the importance of network monitoring within any organisation, let's have a look at some of the reasons for this:

1. *Helps us plan for changes:* Network monitoring solutions allow us to study any constant problem in more detail. For instance, if some hardware keeps constantly tripping, we may need to replace it. The same applies to a service that crashes repeatedly.
2. *Keeps you informed:* With a real-time monitoring system, if any failure or irregularity is detected, it can be immediately communicated by different means such as SMS, emails, pagers or a network message. Hence, we will be notified of any problem on our network, wherever we may be, which allows us to fix the issue swiftly. Without any network monitoring solution, we would have to look for issues on our own, or wait for the issue to be reported to us, to work towards a solution.
3. *Reports issues:* Network monitoring reports can help us spot specific trends in the system's performance. These highlight the need for any upgrades or replacements, and document the otherwise 'unseen' work which keeps the IT systems we manage, running smoothly.
4. *Diagnoses issues:* Imagine a scenario where one of your company's websites goes down. Without network monitoring, you may not be able to even tell if the problem is related to just the website, the Web server or the applications on which the website runs. Network monitoring will actually pinpoint the specific point of failure, saving your time and money, which you would otherwise have had to spend to diagnose the problem.
5. *Remedies disasters:* If you are immediately notified that there is some issue with one of your systems on a network, and the issue might take quite some time to fix, then the time saved by being alerted immediately can be actually used to bring in a backup system that can replace the current failure, thereby providing an efficient service to your customers. Some network monitoring solutions can even automatically move to correct the problem caused by restarting a service (or multiple services) upon failure.
6. *Keeps track of your Web application:* Many services that companies offer to their users or customers are actually just Web applications running on a server. Network monitoring solutions allow you to stay on top of different website problems, spot issues before users or customers notice them, and remedy those issues in a timely fashion.
7. *Ensures the efficient operation of security systems:* Although businesses spend a lot of money, resources and time on security hardware or software, without a network monitoring solution, they cannot be really sure that the security devices are working as expected. Network monitoring solutions can effectively monitor and manage the health of such critical software and hardware security systems. With the help of another feature that this product offers, i.e., patch management, we can also streamline the automation and management of different Microsoft software updates and patches.
8. *Fixes problems, anytime, anywhere:* Nowadays, network monitoring products are being shipped with different remote access features. They offer just one-click remote support for any server or workstation in your environment. Apart from providing a much faster service, remote access also helps in saving a lot of money without having to roam to branch offices or customer sites.
9. *Saves money:* Network monitoring products help in fixing issues faster with instant alerts, spot small and big issues, and eliminate the need for any manual checks on different event logs, backup systems, hard disks, antivirus systems and other devices. All this facilitates cost saving as well as revenue building.
10. *Ensures uptime:* Network monitoring maximises network availability by monitoring all systems on your network, including workstations, servers and network devices or applications. Whenever a failure is detected, you will immediately be notified via the alerts that you configure in the product, allowing you to take corrective action in a highly efficient manner.

Monitoring security aspects of the network

It's so important for IT administrators to be able to react as quickly as possible in order to protect a system from potential malware attacks. If installed antivirus systems and firewalls don't discover these attacks in time, then the damage done can even bring all operations to a standstill. At that time, administrators will just be able to react to these problems, instead of being able to proactively take measures to prevent these problems before they occur. The fact is that these firewalls and virus scanners alone are not always sufficient to ensure the all-around security of the network. Companies that integrate a network monitoring solution in their security strategy are able to discover these potential dangers to the company network at early stages in the following ways.

1. Network monitoring solutions help to check the existing security systems, such as firewalls and virus scanners, for reliability. For example, the monitoring solution gathers detailed data regarding the performance as well as status of the firewall, around the clock. If the firewall is not working properly, then the risk of a malware attack on the network becomes high. To avoid this, the administrators are informed of abnormalities in the firewall at the early stages.
2. The monitoring software also checks different virus scanners running on the central mail server. This helps different companies to make sure that the scanner is continuously active. The monitoring solution even uses special sensors to check the Windows Security Center in

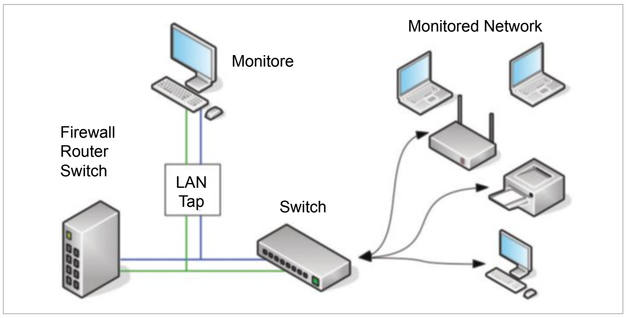


Figure 1: Network monitoring ensures all-round security for networks using firewalls (*Image credits: www.paessler.com*)

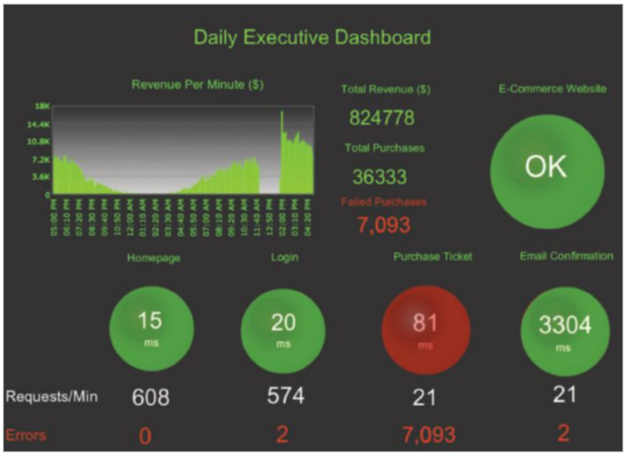


Figure 2: Sample analysis report of the Hypervic tool (Image credits: *http://www.computerworlduk.com/*)

order to ensure that the virus scanners and different anti-malware programs on each computer within the company are up-to-date. This ensures that the client computers are continuously protected against any malware as well.

3. Network monitoring solutions help the administrator measure the bandwidth for leased lines, devices (routers, switches), network connections, etc. Detailed monitoring of the bandwidth usage can also indirectly detect malware attacks. An indication of such an attack may be slow response times from different applications and websites, caused by a malware program that actually eats up a large amount of the bandwidth.

A few open source network monitoring tools

Here are some of the open source network monitoring tools widely available in the market.

Hyperic: This has originated at VMware. It has been developed for monitoring different custom Web applications and their performance across all physical, virtual and cloud environments. Hypervic works across Web servers, application servers, databases, operating systems, messaging servers, hypervisors and directory servers. This network monitoring tool offers an enterprise version that helps in improving the alerting functions and is also able to create better baselines.

Functionality and usage

It helps in:

- OS monitoring
- Detailed reporting
- Application and middleware monitoring
- Alerts and remediation workflows

Zenoss Core: This is another open source stalwart which gives network administrators a complete solution for tracking and managing various applications, servers, networking components, storage, virtualisation tools, etc. Administrators can make sure that the hardware is running efficiently and they can even take advantage of the modular design to plug in different ZenPacks for extended functionality. Zenoss Core 5 was released in February 2016 in order to improve the already powerful tool with an enhanced user interface and an expanded dashboard. It's Web-based console and dashboards were already highly dynamic and customisable. The new version now helps administrators mash up various components' charts onto a single chart. Hence, it's actually the tool for better root cause analysis.

Functionality and usage

It helps in:

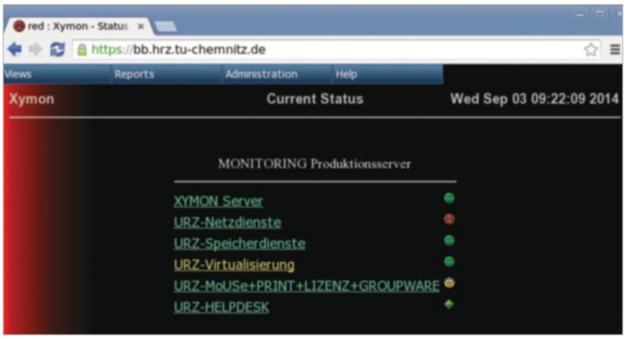
- Network mapping
- Monitoring device issues, daemon processes and production states by listing different event views
- Out-of-band management and monitoring of all Zenoss components
- Online backup, restore, snapshots and multi-host deployment

Xymon: This is a significant network monitoring tool which was formerly known as Hobbit. It was developed to address the shortcomings of tools like Big Brother and Big Sister. It's actually very easy to deploy Xymon on any system and it is, of course, available free of cost.

Functionality and usage

It helps in:

- Monitoring servers, applications and networks
- Offering information about the health of the various components networked via Web pages

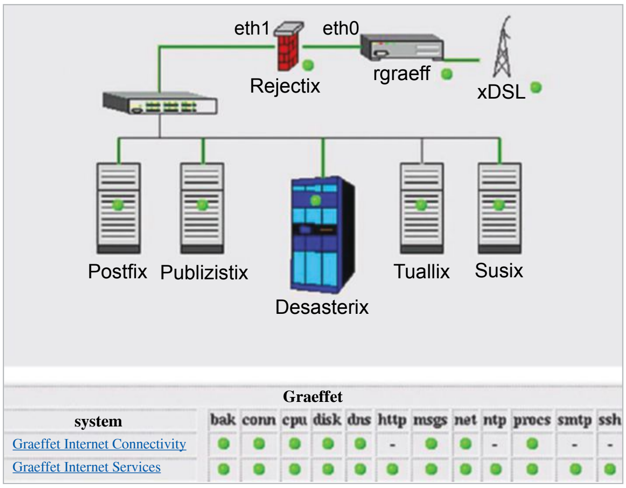


Xymon   Current Status   Wed Sep 03 09:22:09 2014

MONITORING Produktionsserver

XYMON Server ●
URZ-Netzdienste ●
URZ-Speicherdienste ●
URZ-Virtualisierung ●
URZ-MoUSe+PRINT+LIZENZ+GROUPWARE ●
URZ-HELPDESK ◆

Figure 3: Status report of the Xymon tool (Image credits: *http://www.computerworlduk.com/*)



	Graeffet											
system	bak	conn	cpu	disk	dns	http	msgs	net	ntp	procs	smtp	ssh
Graeffet Internet Connectivity	●	●	●	●	●	-	●	●	-	●	-	-
Graeffet Internet Services	●	●	●	●	●	●	●	●	●	●	●	●

Figure 4: Part of an analysis done by the Big Sister tool (Image credits: *http://www.computerworlduk.com/*)

Security Onion: We should all be aware that network security monitoring is made up of many layers, just like an onion. Hence, no single tool will give us visibility into each and every attack or show us every reconnaissance or foot-print session on our company network. Security Onion actually bundles scores of different proven tools into one handy Ubuntu distro that allows us to see who is inside our network and helps keep the bad ones out. Whether we are taking a proactive approach to network security monitoring or even if we are following up on an attack, Security Onion can assist us.

Consisting of server, sensor and display layers, Security Onion combines full network packet capture with network-based and host-based intrusion detection. The network security toolchain also includes Netsniff-NG for packet capture; Suricata and Snort for rules-based network intrusion detection; OSSEC for host intrusion detection; Bro for analysis-based network monitoring; and Sguil, Snorby, Squert and ELSA (Enterprise Log Search and Archive) for display, analysis and log management. It's actually a collection of tools, all wrapped into a wizard-driven installer and backed by thorough documentation that can help us get complete network monitoring as fast as possible.

Functionality and usage

It helps in:

- Combining full network packet capture with the network-based and host-based intrusion detection
- Serving all different logs for inspection and analysis
- Analysis-based network monitoring
- Log management

Big Sister: This tool was created by Thomas Abey, since he was really impressed by the network monitoring functions performed by another such tool called Big Brother. Thomas wanted to improve the performance of the tool and reduce the number of alarms when some system goes down, while making other enhancements. Big Sister uses Node Director, Deoxygen Filter and Big Sister Web application frameworks in order to work as part of different UNIX derivatives and Microsoft Windows versions.

Functionality and usage

It helps in:

- Notifying admins when the system is in a critical state
- Generating history of status changes and logs
- Displaying a variety of system performance data

Benefits of network monitoring

Let's look at the benefits that organisations get out of network monitoring and management.

1. It helps in optimising the availability and performance of any network that is being monitored.
2. Network monitoring also helps in lowering the expenses of any organisation implementing it by improving asset utilisation.
3. It minimises the risks associated with the whole system by providing a secure network which meets compliance guidelines
4. Monitoring of a network also ensures effective change management so that users can establish the solid baselines for its performance.
5. With optimal asset utilisation achieved by network monitoring, the terms of service level agreements are met and it is possible to document the performance using reports. END

References

Network Monitoring and Management by Esad Saitović and Ivan Ivanović.

[1] *http://www.wikipedia.org/*

[2] *http://www.computerworlduk.com/*

[3] *http://www.infoworld.com/*

By: Vivek Ratan

The author currently works as an automation test engineer at Infosys, Pune. He can be reached at *ratanvivek14@gmail.com*.

The Best Open Source Network Intrusion Detection Tools

In enterprises, preventing breaches in the network in order to protect data is a serious matter. Any malware exploit can cost the company a lot. Maintaining networks securely is an aim that all systems administrators hope to achieve. Let us take a look at a few important open source network intrusion detection tools.

In today’s world, data breaches, threats, attacks and intrusions are becoming highly sophisticated. Cyber criminals and hackers come up with new methods of gaining access to business and home networks, making a multi-tiered approach to network security an urgent necessity. An Intrusion Detection System (IDS) is, therefore, the most important tool to be deployed to defend the network against the high tech attacks that emerge daily. An IDS, which is a network security tool, is built to detect vulnerability exploits against a target application or computer. It is regarded as a high-end network device or software application that assists the network or systems administrators in monitoring the network or system for all sorts of malicious activities or threats. Any unusual activity is reported to the administrator using a security information and event management (SIEM) system.

There are a wide variety of IDSs available, ranging from antivirus to hierarchical systems, which monitor network traffic. The most common ones are listed below.

- ***NIDS:*** Network intrusion detection systems are placed at highly strategic points within the network to monitor inbound and outbound traffic from all devices in the network. But scanning all traffic could lead to the creation of bottlenecks, which impacts the overall speed of the network.
- ***HIDS:*** Host intrusion detection systems run on separate machines or devices in the network, and provide safeguards to the overall network against threats coming from the outside world.
- ***Signature based IDS:*** Signature based IDS systems monitor all the packets in the network and compare them against the database of signatures, which are pre-configured and pre-determined attack patterns. They work similar to antivirus software.
- ***Anomaly based IDS:*** This IDS monitors network traffic and compares it against an established baseline. The baseline determines what is considered *normal* for the network in terms of bandwidth, protocols, ports and other devices, and the IDS alerts the administrator against all sorts of unusual activity.
- ***Passive IDS:*** This IDS system does the simple job of detection and alerting. It just alerts the administrator for any kind of threat and blocks the concerned activity as a preventive measure.

- ***Reactive IDS:*** This detects malicious activity, alerts the administrator of the threats and also responds to those threats.

Numerous open source tools are available for enterprise networks, depending on the level of sophistication and security desired. In order to make the network highly secure, an IDS/IPS system should detect all sorts of suspicious activities coming to/from hosts in the network, and should take combative measures to prevent the attack.

Top 8 open source network intrusion detection tools

Here is a list of the top 8 open source network intrusion detection tools with a brief description of each.

Snort

Snort is a free and open source network intrusion detection and prevention tool. It was created by Martin Roesch in 1998. The main advantage of using Snort is its capability to perform real-time traffic analysis and packet logging on networks. With the functionality of protocol analysis, content searching and various pre-processors, Snort is widely accepted as a tool for detecting varied worms, exploits, port scanning and other malicious threats. It can be configured in three main modes -- sniffer, packet logger and network intrusion detection. In sniffer mode, the program will just read packets and display the information on the console. In packet logger mode, the packets will be logged on the disk. In intrusion detection mode, the program will monitor real-time traffic and compare it with the rules defined by the user.

Snort can detect varied attacks like a buffer overflow, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts, etc. It is supported on a number of hardware platforms and operating systems like Linux, OpenBSD, FreeBSD, Solaris, HP-UX, MacOS, Windows, etc.

Pros:

- Free to download and is open source.
- Easy to write rules for intrusion detection.
- Highly flexible and dynamic in terms of live deployments.
- Good community support for solving problems and is under rapid development.

Cons:

- No GUI interface for rule manipulation.
- Somewhat slow in processing network packets.
- Cannot detect a signature split over multiple TCP packets, which occurs when packets are configured in *inline* mode.

Latest version: 2.9.9.0
Official website: *www.snort.org*

Security Onion

Security Onion is a Linux distribution for intrusion detection, network security monitoring and log management. The open source distribution is based on Ubuntu and comprises lots of IDS tools like Snort, Suricata, Bro, Sguil, Squert, Snorby, ELSA, Xplico, NetworkMiner, and many others. Security Onion provides high visibility and context to network traffic, alerts and suspicious activities. But it requires proper management by the systems administrator to review alerts, monitor network activity and to regularly update the IDS based detection rules.

Security Onion has three core functions:

- Full packet capture
- Network based and host based intrusion detection systems
- Powerful analysis tools

Full packet capture: This is done using netsniff-ng, which captures all network traffic that Security Onion can see, and stores as much as your storage solution can hold. It is like a real-time camera for networks, and provides all the evidence of the threats and malicious activities happening over the network.

Network-based and host-based IDS: It analyses the network or host systems, and provides log and alert data for detected events and activity. Security Onion has varied IDS options like rule-driven IDS, analysis-driven IDS, HIDS, etc.

Analysis tools: In addition to network data capture, Security Onion comprises various tools like Sguil, Squert, ELSA, etc, for assisting administrators in analysis.

Security Onion also provides diverse ways for the live deployment of regular standalone, server-sensor and hybrid monitoring tools.

Pros:

- Provides a highly flexible environment for users to tune up network security as per the requirements.
- Consists of pre-installed sensor management tools, traffic analysers and packet sniffers, and can be operated without any additional IDS/IPS software.
- Has regular updates to improve security levels.

Cons:

- Doesn't work as an IPS after installation, but only as an IDS, and the user cannot find any instructions regarding this on the website.
- Doesn't support Wi-Fi for managing the network.
- Additional requirement for admins to learn various tools to make efficient use of the Security Onion distribution.
- No automatic backups of configuration files except rules; so usage of third party software is required for this activity.

Latest version: 14.04.5.1
Official website: *https://securityonion.net/*

OpenWIPS-NG

OpenWIPS-NG is a free wireless intrusion detection and prevention system that relies on sensors, servers and interfaces. It basically runs on commodity hardware. It was developed by Thomas d'Otrepe de Bouvette, the creator of Aircrack software. OpenWIPS uses many functions and services built into Aircrack-NG for scanning, detection and intrusion prevention.

The three main parts of OpenWIPS-NG are listed below.

Sensor: Acts as a device for capturing wireless traffic and sending the data back to the server for further analysis. The sensor also plays an important role in responding to all sorts of network attacks.

Server: Performs the role of aggregation of data from all sensors, analyses the data and responds to attacks. Additionally, it logs any type of attack and alerts the administrator.

Interface: The GUI manages the server and displays the information regarding all sorts of threats against the network.

Pros

- Modular and plugin based.
- Software and hardware required can be built by DIYers.
- Additional features are supported via use of plugins.

Cons

- Only works for wireless networks.
- Only suitable for low and medium level administration, and not fully compliant for detecting all sorts of wireless attacks.
- No detailed documentation and community support compared to other systems.

Latest version: OpenWIPS-NG 0.1 beta 1
Official website: *http://www.openwips-ng.org/*

Suricata

Suricata is an open source, fast and highly robust network intrusion detection system developed by the Open Information Security Foundation. The Suricata engine is capable of real-time intrusion detection, inline intrusion prevention and network security monitoring. Suricata consists of a few modules like Capturing, Collection, Decoding, Detection and Output. It captures traffic passing in one flow before decoding, which is highly optimal. But unlike Snort, it configures separate flows after capturing and specifying how the flow will separate between processors.

Pros:

- Does the network traffic processing on the seventh layer of the OSI model which, in turn, enhances its capability to detect malware activities.
- Automatically detects and parses protocols like IP, TCP, UDP, ICMP, HTTP, TLS, FTP, SMB and FTP so that rules apply on all protocols.
- Advanced features consist of multi-threading and GPU acceleration.

Cons:

- Less support as compared to other IDSs like Snort.
- Complicated in operation and requires more system resources for full-fledged functioning.

Latest version: 3.2
Official website: *https://suricata-ids.org*

BroIDS

BroIDS is a passive, open source network traffic analyser developed by Vern Paxson, and is used for collecting network measurements, conducting forensic investigations, traffic base lining and much more. BroIDS comprises a set of log files to record network activities like HTTP sessions with URIs, key headers, MIME types, server responses, DNS requests, SSL certificates, SMTP sessions, etc. In addition, it provides sophisticated functionality for the analysis and detection of threats, extracting files from HTTP sessions, sophisticated malware detection, software vulnerabilities, SSH brute force attacks and validating SSL certificate chains.

BroIDS is divided into the following two layers.

Bro Event Engine: This does the task of analysing live or recorded network traffic packs using C++ to generate events when something unusual happens on the network.

Bro Policy Scripts: These analyse events to create policies for action, and events are handled using policy scripts such as sending emails, raising alerts, executing system commands and even calling emergency numbers.

Latest version: Bro 2.5
Official website: *www.bro.org*

Pros:

- Highly flexible as BroIDS uses a scripting language to allow users to set monitoring rules for each protected object.
- Works efficiently in networks with large volumes of traffic and handles big network projects.
- Capable of in-depth analysis of traffic and supports analysers for multiple protocols. Highly stateful and does forensic level comprehensive log maintenance.

Cons:

- Not easy to handle as it has a complex architecture.
- Programming experience is required for competent handling of the BroIDS system.

OSSEC

OSSEC is a free and open source host based IDS that performs varied tasks like log analysis, integrity checking, Windows registry monitoring, rootkit detection, time-based alerting and active response. The OSSEC system is equipped with a centralised and cross-platform architecture allowing multiple systems to be accurately monitored by administrators.

The OSSEC system comprises the following three main components.

- **Main application:** This is a prime requirement for installations; OSSEC is supported by Linux, Windows, Solaris and Mac environments.
- **Windows agent:** This is only required when OSSEC is to be installed on Windows based computers/clients as well as servers.
- **Web interface:** Web based GUI application for defining rules and network monitoring.

Pros:

- Multi-platform IDS system providing real-time and configurable alerts.
- Centralised management, with agent and agentless monitoring.
- Can be used both in serverless and server-agent mode.

Cons:

- Upgrade process overwrites existing rules with out-of-the-box rules.
- Pre-sharing keys can be troublesome.
- Windows OS is only supported in server-agent mode.

Latest version: 2.8.3
Official website: *http://ossec.github.io/*

Open Source Tripwire

Open Source Tripwire is a host based intrusion detection system focusing on detecting changes in file system objects. On the first initialisation, Tripwire scans the file system as instructed by the systems administrator and stores the information of each file in a database. When files are changed and on future scans, the results are compared with the stored values and changes are reported to users.

Tripwire makes use of cryptographic hashes to detect changes in files. In addition to scanning file changes, it is used for integrity assurance, change management and policy compliance.

Pros:

- Excellent for small, decentralised Linux systems.
- Good integration with Linux.

Cons:

- Only runs on Linux.
- Requires the user to be a Linux expert.
- Advanced features are not available in open source versions.
- No real-time alerts.

Latest version: 2.4.3.1
Official website: *https://github.com/Tripwire/tripwire-open-source*

AIDE

AIDE (Advanced Intrusion Detection Environment) was developed by Rami Lehti and Pablo Virolainen. It is regarded as one of the most powerful tools for monitoring changes to UNIX or Linux systems. AIDE creates a database via regular expression rules that it finds from the *config* files. On initialising the database, it is used to verify the integrity of files.

Some of the most powerful features of AIDE are as follows:

- Supports all kinds of message digest algorithms like MD5, SHA1, RMD160, TIGER, SHA256 and SHA512.
- Supports POSIX ACL, SELinux, XAttra and Extended File System.
- Powerful regular expression support to include or exclude files and directories for monitoring.
- Supports various operating system platforms like Linux, Solaris, Mac OS X, UNIX, BSD, HP-UX, etc.

Pros:

- Real-time detection and elimination of the attacker to restore file or directory properties.
- Anomaly detection to reduce the false rate of file system monitors.
- Supports a wide range of encryption algorithms.

Cons:

- No GUI interface.
- Requires careful configuration for effective detection and prevention.
- Doesn't deal properly with long file names for smooth detection.

Latest version: 0.16
Official website: *http://aide.sourceforge.net/* END

References

[1] *www.snort.org*
[2] *https://securityonion.net/*
[3] *http://www.openwips-ng.org/*
[4] *https://suricata-ids.org*
[5] *www.bro.org*
[6] *http://ossec.github.io/*
[7] *https://github.com/Tripwire/tripwire-open-source*
[8] *http://aide.sourceforge.net/*

By: Prof. Anand Nayyar

The author is an assistant professor in the department of computer applications and IT at KCL Institute of Management and Technology, Jalandhar, Punjab. He loves to work on and research open source technologies, cloud computing, sensor networks, hacking and network security. He can be reached at *anand_nayyar@yahoo.co.in.* Watch his YouTube videos at *Youtube.com/anandnayyar.*

Identifying and Mitigating Distributed Denial of Service Attacks

A distributed denial of service (DDoS) attack involves the paralysing of a network by flooding it with data from several individual sources, to the detriment of genuine users. The identification and mitigation of such attacks is an important issue for systems administrators.

With the advent and adoption of technology-loaded devices, a number of challenges are regularly resolved by cyber security professionals. These include breach of privacy, data sniffing and theft, integrity, access control, fake traffic, channel damage, and many others. Cyber security experts are continuously working to enforce and integrate a higher level of security in the devices as well as the network infrastructure so that users can access and use the technology without any vulnerability issues.

In a technology based environment, the hacking community regularly launches numerous attacks. A number of algorithms and mechanisms have been devised to cope with such virtual attacks. Research is still under way in the domain of securing applications, devices, networks and computing infrastructure.

Broadly, there are two types of attacks in any network based environment.

Passive attacks: In any network environment, when there is an attempt at sniffing the data channels to copy secret information, it is considered to be a passive attack. In such cases, the modification of files, directories or credentials is not done. Passive attacks are primarily used to monitor the remote system or server, to spy on private information.

Active attacks: In the case of active attacks, the effect and damage of the assault is instant and, often, the victim machine would not have been prepared for such attacks. These include injection of malicious traffic to damage the remote system, updating or deleting remote files, modifying authentication files and much more.

Distributed denial of service (DDoS) attack

A distributed denial of service (DDoS) attack is a powerful assault, in the taxonomy of active attacks, which is used to restrict access to services from authenticated users. In DDoS attacks, genuine users are not allowed to use a system or service because of excessive traffic.

In very simple terms, fake or malicious traffic is generated in large volumes on a server in order to overload it, thus resulting in the network getting choked or jammed. DDoS attacks are often known as jamming attacks because fake traffic or data packets overload the server delivering a particular service. Due to this attack, other users are not allowed to access the service because of massive congestion in the network.

As an example, let's suppose there is a limit of 200 concurrent users who can access a website. In a DDoS attack, 200 sources of website access can be generated. After those 200 connections are captured by fake traffic, the genuine users will not be able to access that particular website because of traffic overload or channel congestions.

Types of DDoS attacks

- *Application layer based DDoS*: Such an attack is used to target and damage the application layer of the network. The effect of this attack is measured in terms of requests per second (RPS). A large number of RPS are generated, which becomes a load for the network.
- *Protocol based DDoS*: In this type of attack, the resources and related modules of the server are the victims. Bandwidth is not captured in protocol based DDoS.
- *Volume based DDoS*: Bandwidth is the key target here: it is saturated and flooded with massive traffic in volume based DDoS attacks. If such an attack is successful, the server crashes and major flaws occur.

HULK based DDoS

Whenever there is a DDoS attack on a website, it is known

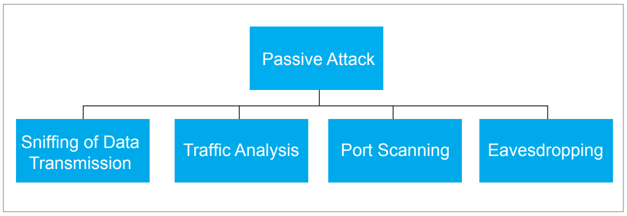


Figure 1: Passive attacks in the network environment

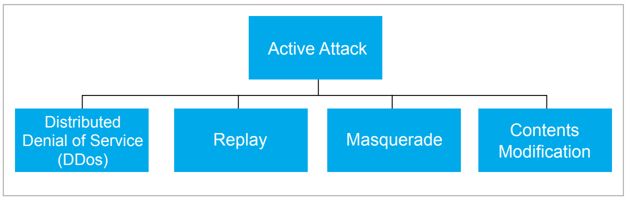


Figure 2: Active attacks in the network environment

as a HULK (HTTP unbearable load king) attack. In a HULK attack, the unbearable load is created at the HTTP service. A number of virtual connections are created and then fired at the website. If a HULK attack is used, the particular website gets a large number of connections from fake traffic and then the website hangs. That's why HULK is classified as a DDoS attack. HULK attacks are generally carried out using Python, PHP, Java or Perl scripts, which are easily available on assorted Web based repositories of source code.

As there are massive DDoS attacks on different websites and servers, it is mandatory for network administrators to adopt and implement mechanisms to cope with and mitigate such attacks.

Let's now discuss two free and open source tools that help to detect and repel such DDoS attacks.

DDoS Deflate

DDoS Deflate, an open source tool, is a powerful shell script to cope with DDoS attacks on servers. DDoS Deflate is dominant enough to push back and block DDoS attacks. At the base level, it makes use of the *netstat* command to identify and investigate the IP addresses that are creating connections with the server.

The following command is used to identify and list the connections created by all the IP addresses:

```
<workingdirectory>$ netstat -ntu | awk '{print $5}' | cut -d:
-f1 | sort | uniq -c | sort -n
```

The features of DDoS Deflate include:

- Auto blocking of IP addresses
- Blacklisting and whitelisting of traffic and their sources
- Easy notification and management for network administrators
- Auto detection of rules associated with Iptables and advanced policy firewalls
- Ease in configuration
- Auto e-mail alerts
- Uses *tcpkill* to push back the unwanted and fake connections

Fire the following commands in a terminal to install DDoS Deflate:

```
<workingdirectory>$ cd /usr/local/src/
<workingdirectory>$ wget http://www.inetbase.com/scripts/
ddos/install.sh
<workingdirectory>$ chmod 0700 install.sh
<workingdirectory>$ ./install.sh
```

The configuration file of DDoS Deflate can be edited as follows:

```
<workingdirectory>$ vi /usr/local/ddos/ddos.conf
```

or

```
<workingdirectory>$ gedit /usr/local/ddos/ddos.conf
```

Use the following command to start DDoS Deflate:

```
<workingdirectory>$ /usr/local/ddos/ddos.sh –c
```

The following code will uninstall DDOS Deflate:

```
<workingdirectory>$ wget http://www.inetbase.com/scripts/
ddos/uninstall.ddos
<workingdirectory>$ chmod 0700 uninstall.ddos
<workingdirectory>$ ./uninstall.ddos
```

To view the 'Help' screen and all other options in DDoS Deflate, type:

```
<workingdirectory>$ ddos –help
```

To view whitelisted IP addresses, type:

```
<workingdirectory> $ ddos -I | –ignore-list
```

Use the following command to view banned or blacklisted IP addresses:

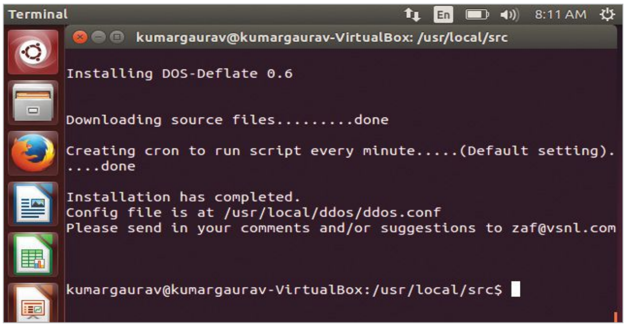
```
<workingdirectory> $ ddos -b | –bans-list
```

To initialise the daemon process for monitoring connections, type:

```
<workingdirectory> $ ddos -d | –start:
```

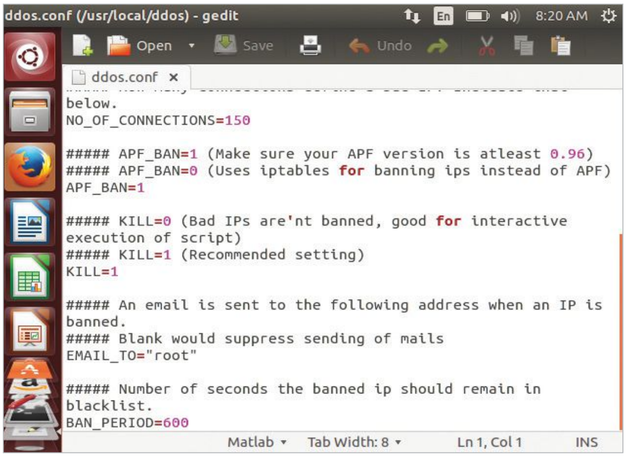
Type the following command to stop the daemon process:

```
<workingdirectory>$ ddos -s | –stop
```



```
Installing DOS-Deflate 0.6

Downloading source files.........done

Creating cron to run script every minute.....(Default setting).
....done

Installation has completed.
Config file is at /usr/local/ddos/ddos.conf
Please send in your comments and/or suggestions to zaf@vsnl.com

kumargaurav@kumargaurav-VirtualBox:/usr/local/src$
```

Figure 3: DDoS Deflate working environment



```
below.
NO_OF_CONNECTIONS=150

##### APF_BAN=1 (Make sure your APF version is atleast 0.96)
##### APF_BAN=0 (Uses iptables for banning ips instead of APF)
APF_BAN=1

##### KILL=0 (Bad IPs are'nt banned, good for interactive
execution of script)
##### KILL=1 (Recommended setting)
KILL=1

##### An email is sent to the following address when an IP is
banned.
##### Blank would suppress sending of mails
EMAIL_TO="root"

##### Number of seconds the banned ip should remain in
blacklist.
BAN_PERIOD=600
```

Figure 4: Configuration file of DDoS Deflate

To view the current status of the daemon and PID running, give the following command:

```
<workingdirectory>$ ddos -t | -status
```

You can view the active connections with the server by typing the following command:

```
<workingdirectory>$ ddos -v | -view
```

The following command bans or blacklists all IP addresses with more than ‘n’ connections:

```
<workingdirectory>$ ddos -k | -kill:
```

Fail2Ban

Fail2Ban is another free and open source tool to identify and ban the sources of malicious DDoS traffic. It scans the log files and identifies suspicious patterns and connections so that blacklisting can be done. Fail2Ban reduces the non-legitimate and incorrect authentication attempts with the use of powerful modules for filtering the various services.

The features of Fail2Ban include:

- Deep parsing and analysis of log files
- Awareness of the time zone associated with the source traffic IP
- Integration of client-server architecture
- Assorted services including sshd, vsftpd and Apache can be processed
- Easy configuration for the administrator
- Compatibility with all the firewalls
- Whitelisting and banning of IP addresses
- Blocking of brute force assaults
- Blocking of IP addresses based on time slots
- Excellent for SSH based environments
- Time-based IP blocking
- Support for Python programming

To install and work with Fail2Ban, type the following command:

```
<workingdirectory>$ sudo apt-get install fail2ban
```

Fail2Ban service maintains a configuration file in the directory */etc/fail2ban*. In this directory, the default configuration file is *jail.conf*.

After installation, the default configuration file is copied to the working configuration file.

```
<workingdirectory>$ sudo cp /etc/fail2ban/jail.conf /etc/
fail2ban/jail.local
```

The configuration settings at the end of the *config* file are as follows:

```
[http-get-dos] # Rule to be Set
enabled = true   # Status
port = http,https # 80,443 (Ports)
filter = http-get-dos # Filter Names
logpath = /var/log/www/vhost.d/mysite.com/site-access_log #
Path of Log
maxretry = 5   # Retries Max. Limit
findtime = 10   # 5 retries in 10 seconds from 1 IP Ban or
Blacklist
bantime = 86400 # In Seconds (One Day)
action = iptables[name=HTTP, port=http, protocol=tcp]
             iptables[name=HTTPS, port=https, protocol=tcp]
             sendmail-whois-withline[name=httpd-get-dos,
dest=<E-mail ID>, logpath=/var/log/httpd/site-access_log] #
sets iptables variables.
```

To view all the *jail* files that have been enabled, type:

```
<workingdirectory>$ sudo fail2ban-client status
```

END

By: Dr Gaurav Kumar

The author is the MD of Magma Research and Consultancy Pvt Ltd. He is associated with various academic and research institutes, where he delivers expert lectures and conducts technical workshops on the latest technologies and tools. He can be contacted at *kumargaurav.in@gmail.com*.

"The community drives OpenStack's innovation curve"

In the world of open source, OpenStack has emerged as the ultimate solution to manage the cloud. It is an alternative to all the existing proprietary cloud management platforms and is considered to be easier to deploy than traditional offerings. But what are the prime advantages of opting for OpenStack in today's fast-moving IT world? ***Mark Collier, co-founder*** and ***COO, OpenStack Foundation***, describes those advantages in an exclusive conversation with ***Jagmeet Singh*** of ***OSFY.***

Q What was the original idea behind developing OpenStack?
OpenStack was launched in 2010 through the initial contributions of Rackspace (a Swift Storage project) and NASA (the Nova Compute Project) as an open source cloud infrastructure management project. It quickly grew to include projects for block storage, networking, authentication, dashboards, container management and much more. Today, OpenStack is in its 14th release, called Newton. It is in production use worldwide by hundreds of companies, supporting a wide diversity of applications, and has emerged as the *de facto* private cloud standard as well as a widely deployed public and hybrid cloud platform.

Q What factors helped Rackspace and NASA to jointly bring the OpenStack movement into the open source ecosystem?
One defining characteristic of the two founding organisations was the lack of any desire to directly monetise the software through an 'enterprise' version. Neither were software companies, per se. Instead, they simply wanted to see the technology develop rapidly. This helped build a large and thriving ecosystem fast, because it was easier to establish the trust that no one company would dominate the stack.

Q How is this OpenStack movement different from the initial development of Linux?
The biggest difference between Linux and OpenStack lies in the way each one's community is structured. Linux is guided almost exclusively by its creator, Linus Torvalds. It is a model that has worked really well for Linux. OpenStack, by contrast, is governed by a technical committee and board of directors, who are guided and informed by an active and engaged group of developers and users from a diverse collection of companies and organisations. It is a model that is not without challenges but has on balance worked well for open source cloud management and is a prime factor in the success of the project.

Mark Collier, co-founder and COO, OpenStack Foundation

Q What is your opinion on Microsoft Azure, Amazon Web Services and Google Cloud? How is OpenStack distinct from these major cloud platforms?
The major difference, of course, is that OpenStack is open source and capable of deployment both as a service and on-premise, while the others are proprietary and only available as a service—at least for the moment. OpenStack works with each of these platforms in a hybrid cloud context, either through APIs or via tighter degrees of integration. Vendors in our ecosystem have developed a variety of products and services to complement OpenStack with these public cloud providers, making the hybrid cloud real and productive for a growing number of users.

Q Why should one opt for OpenStack over VMware or any other proprietary cloud platform?
OpenStack is an open source cloud management solution,

which means you control your own destiny, with the support of a wide community rather than a single service provider. Many users want that kind of assurance underpinning their cloud strategies.

Q How does OpenStack ease the transition from a public cloud to a private or hybrid one?

The OpenStack community has worked hard to simplify these transitions with projects like the Interoperability Challenge. Also, OpenStack has emerged as the leader with its renewed focus on transitioning from the public cloud to private and hybrid clouds as public cloud deployment sizes grow to the point that they become economically unviable.

Q What are your views on the recent open source adoptions by Microsoft?

It is great to see a growing group of legacy IT players like Microsoft embracing open source. We think vendors who make responsible, participative open source a part of their product roadmaps and go-to-market strategies will achieve better results for their customers compared with those who cling to the proprietary-only model.

Q Do you consider Microsoft's Azure stack as a close competitor to OpenStack?

It is hard to say since it is still in development.

Q What are your plans to ensure the continuous popularity of OpenStack in the cloud world?

The community drives OpenStack's innovation curve. You will notice the growth in the popularity as well as the functionality of OpenStack through any of its developer-focused summits.

Q How can OpenStack play the role of the saviour in this mobile-first world?

Saviour is a big word. Instead, let us look at how OpenStack is solving a very real, existential problem for carriers and operators of big networks that form the backbone of our mobile-first world. That problem is providing reliable, agile networks to support the explosive market growth of mobile devices of all sizes and types. Carriers and service providers in the OpenStack community have used the software to create a powerful solution for network functions virtualisation (NFV) powered by the cloud management platform and technologies like OPNFV. As a result, companies like AT&T are building their networks to run on OpenStack.

A recent survey by Heavy Reading claims that 86 per cent of global telecoms consider OpenStack important to their success.

Q What makes OpenStack the perfect choice for companies like Dell, HP and IBM?

The demand for infrastructure continues to grow, which means more servers, switches and other hardware. OpenStack is the automation software that makes all that infrastructure programmable. As a result, we can say that OpenStack is helping each of the companies that supply the infrastructure market, deliver it in a way that is much easier to manage at scale than ever before.

> “It is great to see a growing group of legacy IT players like Microsoft embracing open source.”

Q Where does India stand in the worldwide adoption of OpenStack?

We do not have precise adoption metrics. However, India is among the top three countries when it comes to our website's traffic. It indicates how important a region the Indian market is for OpenStack.

Q How does the OpenStack Foundation plan to make Indian startups aware of various open source cloud management offerings?

The OpenStack Foundation provides support in a number of ways, including helping the local community organise an 'OpenStack Day' in India each year, and by publishing case studies.

Q What does the OpenStack Foundation offer developers around the globe?

The OpenStack Foundation provides online and location-specific training models for all levels of expertise to enrich developers with enough skills to get work on various cloud developments. Also, we conduct a Certified OpenStack Administrator (COA) exam that is the first professional certification offered by the non-profit body.

Enthusiasts and developers who want to learn about OpenStack and its operations in a cloud premise can catch the latest from *Superuser Magazine*. This online publication has a range of how-tos, case studies and news on the organisational culture.

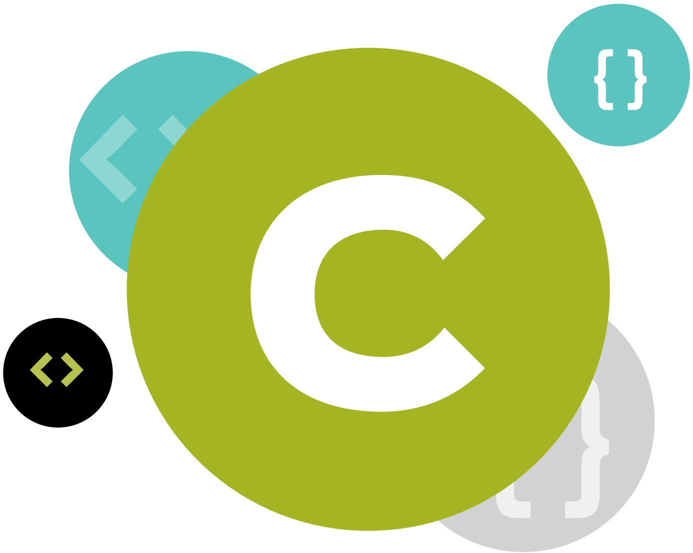
Besides, the OpenStack Foundation maintains Web forums to let developers interact directly with those who operate and develop OpenStack.

Q Do you foresee OpenStack becoming the most dominant solution in the future of cloud computing?

OpenStack will be a major player in a set of technologies and solutions that will drive the next wave of innovation in IT. As OpenStack's current role in developments like NFV and containers shows us, it is just impossible to know what the future holds. That is why the community is focused on creating an agile and flexible platform that can adapt to and support a diversity of innovations in cloud computing and infrastructure management. END

Different C Standards: The Story of C

This article covers the various standards of the C language, from K&R C through ANSI C, C99, C11 and Embedded C.



I have always wanted to write about the different standards of C but refrained from doing so for two reasons. The first is that though as an academician I respect and am devoted to C, I thought the industry hardly cared about it. The other more compelling reason is that not all the different standards of C are taken seriously. So, if nobody cares, readers may wonder why am I bothering to write this article.

Two incidents made me change my mind. To write an article about popular programming languages in *OSFY*, I did a detailed study of programming languages and that included C. I found out that C is still very popular in all the rankings, with supporters in both the academic world and the industry. The other reason is that in a technical interview I had a few months ago, I faced a question based on the C11 standard of C. Even though I absolutely fumbled with that question, I was quite happy and excited. Finally, the times have changed and people are now expected to know the features of C11 and not just ANSI C. So, I believe this is the right time to start preaching about the latest standards of C.

The ANSI C standard was adopted in 1989. In the last 28 years, C has progressed quite a bit and has had three more standards since ANSI C. But remember, ANSI C is not even the first C standard — K&R C holds that distinction. So, there were standards before and after ANSI C. Let's continue with a discussion of all the five different standards of C — K&R C, ANSI C, C99, C11 and Embedded C. For the purposes of our discussion, the compiler used is the *gcc* C compiler from the GNU Compiler Collection (GCC).

If there are five different standards for C, which one is the default standard of *gcc*? The answer is: none of the above. The command *info gcc* will tell you about the current default standard of *gcc*, which can also be obtained with the option *-std=gnu90*. This standard has the whole of ANSI C with some additional GNU-specific features. But there's no need to panic — *gcc* gives you the option to specify a particular standard during compilation, except in the case of K&R C. I will mention these options while discussing each standard.

But if there are so many standards for C, who is responsible for creating a new standard? Can a small group of developers propose a new standard for C? The answer is an emphatic 'No!'. The C standards committee responsible for making the decisions regarding C standardisation is called ISO/IEC JTC 1/SC 22/WG 14. It is a standardisation subcommittee of the Joint Technical Committee ISO/IEC JTC 1 of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).

K&R C

A lot has been written about the history of C and I am not going to repeat it. All I want to say is that C was developed by Dennis Ritchie between 1969 and 1973 at Bell Labs. It was widely accepted by almost all the professional programmers immediately, and they started making their own additions to it. After a few years, there were a lot of C variants available for a programmer to choose from. My C was different from yours, and nobody could claim that theirs was the real one. So, there was a need for standardisation and the masters themselves have undertaken the task.

The standard called K&R C was actually an informal specification based on the first edition of the extremely popular book, ‘The C Programming Language’, published in 1978 by Brian Kernighan and Dennis Ritchie; hence, the name K&R after Kernighan and Ritchie. But do remember that the second edition of the book that is available nowadays covers mostly ANSI C. So, if you want to learn the details of K&R C from the masters themselves, then you have to buy a used copy of the first edition of this book online.

K&R C not only acted as an informal standards specification for C but also added language features like the new data types *long int* and *unsigned int* and the compound assignment operator. A standardised I/O library was also proposed by K&R C. Since the most popular C standard followed by the academia in India is still ANSI C, I will mention a few differences between K&R C and ANSI C. “There are 32 keywords in C,” is one of the clichés I utter in many of my C programming classes. But I often forget to mention the fact that this was not always true. According to K&R C, there are only 28 keywords in C. One keyword was called *entry*, which was neither implemented at that point in time by any of the compilers nor accepted into the list of keywords in ANSI C.

Another major difference is regarding the function definition. A function definition in K&R C has the parameter types declared on separate lines. Consider the following lines of code showing a function definition in K&R C:

```
float fun( a, b )
int a;
float b;
{
      float c;
      c = a * b;
      return c;
}
```

The function *fun( )* accepts an integer and a floating-point variable, and returns the product. You can clearly see that the two parameters are declared on separate lines below the name of the function. This is not the style for function definition in ANSI C. The first line of the same function definition will be *float fun(int a, float b)* in ANSI C. As mentioned earlier, *gcc* does not have an option to specify compilation in the K&R C standard. But programs written in K&R C will also be compiled without any errors, because *gcc* compiler is backward compatible with K&R C.

ANSI C

Even though K&R C was accepted by many programmers as the *de facto* standard of C, it was not the *de jure* standard, and nobody could have been coaxed into accepting it as the official standard of C. So, it was absolutely essential for some standards organisation to accept the challenge of coming up with an official standard for C. The American National Standards Institute (ANSI) addressed this issue in 1983 by forming a committee, and the final draft of the standards it formulated was released in 1989. This is the reason why ANSI C is also called C89.

ANSI C is dependent on the POSIX standard. POSIX is the Portable Operating System Interface, a family of standards specified by the IEEE Computer Society for maintaining compatibility between operating systems. The same standard proposed by ANSI was adopted by ISO officially as ISO/IEC 9899:1990 in 1990. This is the reason why ANSI C is also called as ISO C and C90. One standard and four different names — I believe this the reason why this particular standard of C is still very popular. The following five keywords *const*, *enum*, *signed*, *void* and *volatile* were added, and the keyword *entry* was dropped in ANSI C. The option *-ansi* from *gcc* compiler will compile programs in the ANSI C standard. The options *-std=c89* and *-std=c90* will also compile programs in ANSI C. Since *gcc* is a backward compatible compiler, the above given options will result in successful compilation of programs with K&R C features. Almost all the popular C compilers support ANSI C features. These include compilers like *gcc* from GCC, Portable C Compiler (PCC), *Clang* from LLVM, etc. For examples of ANSI C code, open any textbook on C and you will find a lot of them.

C99

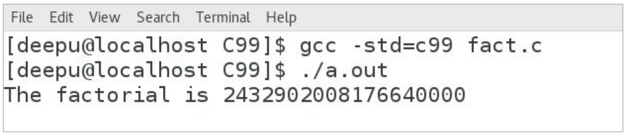
C99 is the informal name given to the ISO/IEC 9899:1999 standards specification for C that was adopted in 1999. The C99 standard added five more keywords to ANSI C, and the total number of keywords became 37. The keywords added in C99 are *_Bool*, *_Complex*, *_Imaginary*, *inline* and *restrict*. The keyword *_Bool* is used to declare a new integer type capable of storing 0 and 1. The keywords *_Complex* and *_Imaginary* are used to declare complex and imaginary floating point type variables to handle complex numbers. The keyword *inline* is used to declare inline functions in C, similar to C++ inline functions. The keyword *restrict* is used to tell the compiler that for the lifetime of the pointer, only the pointer itself or a value directly derived from it will be used to access the object to which it points. New header files like <*stdbool*.h>, <*complex*.h>, <*tgmath*.h>, <*inttypes*.h>, etc, were also added in C99. A new integer data type called *long long int* with a minimum size of 8 bytes was added in C99.

In *gcc* compiler *long long int* usually takes 8 bytes. The C99 program named *fact.c* given below finds the factorial of 20. A program using the data type *long int* with the usual minimum size of 4 bytes would have given the answer as -2102132736 due to overflow.

```
#include<stdio.h>
int main( )
{
     long long int f=1;
     int a=20,i=1;
     for(i=1; i<=a; i++)
     {
       f = f * i;
     }
     printf("the factorial is %lld\n", f);
     return 0;
}
```

The program can be compiled by executing the command *gcc -std=c99 fact.c* or the command *gcc fact.c* on the terminal. The second command works because, by default, *gcc* compiler supports *long long int*. The output of the C99 program is shown in Figure 1.

Features like variable length arrays, better support for IEEE floating point standard, support for C++ style one line comments (//), macros with variable number of arguments, etc, were also added in C99. The official documentation of *gcc* has this to say: '*ISO C99. This standard is substantially completely supported*'. The legal term '*substantially complete*' is slightly confusing but I believe it means that the *gcc* compiler supports almost all the features proposed by the standards documentation of C99. As mentioned earlier, the option *-std=c99* of *gcc* will compile programs in the C99 standard.



```
[deepu@localhost C99]$ gcc -std=c99 fact.c
[deepu@localhost C99]$ ./a.out
The factorial is 2432902008176640000
```

Figure 1: Factorial of 20 in C99

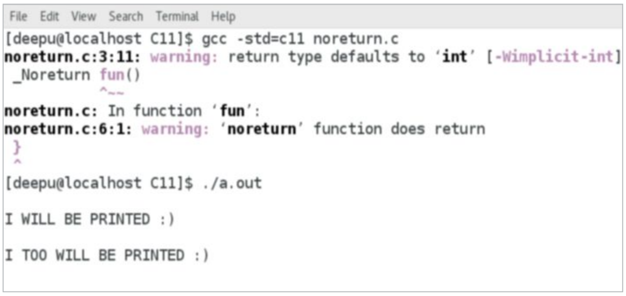
C11

C11 is the current and latest standard of the C programming language and, as the name suggests, this standard was adopted in 2011. The formal document describing the C11 standard is called ISO/IEC 9899:2011. With C11, seven more keywords were added to the C programming language, thereby making the total number of keywords, 44. The seven keywords added to C99 are *_Alignas*, *_Alignof*, *_Atomic*, *_Generic*, *_Noreturn*, *_Static_assert* and *_Thread_local*. Consider the C11 program *noreturn.c* shown below, which uses the keyword *_Noreturn*.

```
#include<stdio.h>

_Noreturn fun( )
{
     printf("\nI TOO WILL BE PRINTED :) \n\n");
}

int main( )
{
     printf("\nI WILL BE PRINTED :) \n");
     fun( );
     printf("\nI WILL NOT BE PRINTED :( WHY? \n");
}
```

Figure 2 shows the output of the program *noreturn.c*. There are a few warnings because the *main( )* has one more line of code after the completion of the function *fun( )*. So why was the last *printf( )* in *main( )* not printed? This is due to the difference between a function returning *void* and a function with the *_Noreturn* attribute. The keyword *void* only means that the function does not return any values to the callee function, but when the called function terminates the program, the program counter register makes sure that the execution continues with the callee function. But in case of a function with the *_Noreturn* attribute, the whole program terminates after the completion of the called function. This is the reason why the statements after the function call of *fun( )* didn't get executed.

The function *gets( )* caused a lot of mayhem and so was deprecated in C99, and completely removed in C11. Header files like *<stdnoreturn.h>* are also added in C11. Support for concurrent programming is another important feature added by the C11 standard. Anonymous structures and unions are also supported in C11. But unlike C99, where the implementation of variable length arrays was mandatory, in C11, this is an optional feature. So even C11-compatible compilers may not have this feature. The official documentation of *gcc* again says that '*ISO C11, the 2011 revision of the ISO C standard. This standard is substantially completely supported*'. So, let us safely assume that the *gcc* compiler supports most of the features proposed by the C11 standard documentation. The option *-std=c11* of *gcc* will compile programs in C11 standard. The C Standards Committee has no immediate plans to come up with a new standard for C. So, I believe we will be using C11 for some time.



```
[deepu@localhost C11]$ gcc -std=c11 noreturn.c
noreturn.c:3:11: warning: return type defaults to 'int' [-Wimplicit-int]
 _Noreturn fun()
           ^~~
noreturn.c: In function 'fun':
noreturn.c:6:1: warning: 'noreturn' function does return
 }
 ^
[deepu@localhost C11]$ ./a.out

I WILL BE PRINTED :)

I TOO WILL BE PRINTED :)
```

Figure 2: *_Noreturn* keyword in C11

Continued on page 74....

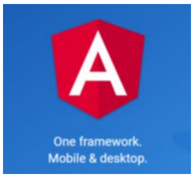
The Latest Open Source Tools for Web Developers

This article aims to acquaint readers with the trendy new open source tools for Web development. These are different from website builders and IDEs as they do not directly assist in the creation of websites. Rather, they are browser add-ons or built into browsers to test the user facing interfaces of the website or Web application.

The Web is growing exponentially nowadays. We have lots of new tools and technologies for rapid Web development, but since we can't include everything in this article, I have put together a set of the latest tools in this domain.

Hopefully, you will find a new tool or resource that will aid you in your Web development workflow.

Angular/AngularJS



AngularJS is a JavaScript-based open source Web application framework for dynamic Web app development. It's mainly maintained by Google, and by a community of individuals and corporations to address many of the challenges encountered in developing single-page applications. It was originally created by Google and open sourced under the MIT licence. Developed by Brat Tech LLC, Google and the community, its initial release was in 2009.

Angular 2 was released on September 14, 2016. This is not a version upgrade, but a complete rewrite of Angular 1.

Angular 2 is a development platform for building mobile and desktop Web applications. It focuses on data-binding, extensible HTML and application test ability, but it is still in the design and prototyping stage.

Its features and benefits are:

- Speed and performance
- Mobile oriented
- Flexible development
- Supports server-side pre-rendering
- Simple and expressive
- Comprehensive routing
- Animations
- Hierarchical dependency injection
- Support for Web components
- Internationalisation, localisation (i18n) and accessibility

Node.js

Node.js is an open source, cross-platform JavaScript runtime environment for developing a diverse variety of tools and applications. It was developed by Ryan Dahl in 2009 and its current stable version is 7.4.0. It's built on Google Chrome's JavaScript Engine (V8 Engine). Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Its package ecosystem, npm, is the largest ecosystem of open source libraries in the world.

Some of the well known users of Node.js are GoDaddy, Groupon, IBM, LinkedIn, Microsoft, Netflix, PayPal, Rakuten, SAP, Voxer, Walmart, Yahoo! and Cisco Systems.

SASS

SASS (syntactically awesome style sheets) is an open source styling language that helps reduce a lot of the repetitive work and maintainability challenges of traditional CSS. It is perhaps the most mature, stable and powerful professional grade CSS extension language in the world.

SASS was initially designed by Hampton Catlin and developed by Natalie Weizenbaum. After its initial versions, Weizenbaum and Chris Eppstein continued to extend SASS with SassScript, a simple scripting language used in SASS files.

SASS is an extension of CSS, adding nested rules, variables, mixins, selector inheritance and more. It is translated to well-formatted, standard CSS using the command line tool or a Web-framework plugin.

Bootstrap

Bootstrap is a free, open source tool well known for the fast development of responsive design. It has a set of its own classes and grids, buttons, forms, navigation, containers, media queries and JavaScript extensions. Bootstrap is the most-starred project on GitHub, with over 91K stars and more than 38K forks.

GitLab

GitLab is an open source tool used by developers to create and manage code bases collaboratively. Built on Git, which is a very popular and efficient distributed version control system, GitLab gives you all the tools needed for Git repository management — from code reviews to issue tracking and more. It is developed by GitLab Inc. The software was written by Dmitriy Zaporozhets and Valery Sizov from Ukraine. Well known users of GitLab are IBM, Sony, Jülich Research Center, NASA, Alibaba, Invincea, O'Reilly Media, Leibniz-Rechenzentrum (LRZ) and CERN.

Its key features are:

- Access to the source code
- Fully modifiable
- Long-term viability
- New stable version released every month
- Built with community support

Elasticsearch

Elasticsearch is one of the most popular, fastest-growing open source search technologies and delivers powerful search. Elasticsearch is a search engine based on Lucene. It provides a distributed, multi-tenant-capable full-text search engine with an HTTP Web interface and schema-free JSON documents. It is developed in Java and is released as open source under the Apache Licence. Elasticsearch is the most popular enterprise search engine followed by Apache Solr (as per the DB-Engines Ranking of Search Engines).

Elasticsearch uses standard RESTful APIs and JSON. It also has built-in clients in many languages such as Java, Python, .NET and Groovy, and a few more contributed by the community.

Elasticsearch was originally developed by Shay Banon. The first version of it was released in February 2010. In March 2015, the company changed its name to Elastic.

Some well-known users of Elasticsearch are Wikimedia, Adobe Systems, Facebook, StumbleUpon, Mozilla, Quora, Foursquare, SoundCloud, GitHub, Stack Exchange, Netflix and many more.

XAMPP

XAMPP is an open source, cross-platform tool which is one of the most favoured by Web developers. The full form of XAMPP is X-Cross platform, Apache, MariaDB, PHP and PERL. Earlier, it used MySQL instead of MariaDB. XAMPP is a complete package of these libraries, so developers don't need to worry about installing and configuring PHP, MariaDB and Apache. It's the simplest way to set up a local Web server.

Notepad++

Notepad++ is an open source text and source code editor for Microsoft Windows. It was developed by Don Ho in September 2003. Notepad++ provides tabbed editing, syntax highlighting and code folding for more than 50 programming, scripting and markup languages. It is distributed as free software. Initially, Notepad++ was hosted on SourceForge.net, from where it has been downloaded over 28 million times. It has twice won the SourceForge Community Choice Award for Best Developer Tool. Since 2015, Notepad++ has been hosted on GitHub. It has wide community support and plugins. Notepad++ also supports Macro recording and playback, Bookmark and PCRE (Perl Compatible Regular Expressions) Search/Replace.

Grunt

Grunt is a JavaScript task runner. It is built on Node.js and is available as a package via the Node package manager (npm).

When you are working on a JavaScript project, there are some tasks that you'll do regularly, like minifying your scripts, compilation, unit testing, running JSHint on your code, etc. You can define this set of tasks in Gruntfile and run a single command from a command line interface to do these tasks. The Grunt ecosystem is huge and it's growing every day. Presently, there are more than five thousand plugins available in the Grunt ecosystem to choose from. You can use Grunt to automate just about anything with minimum effort.

ReactJS

ReactJS is an open source JavaScript library for building user interfaces. It is developed by Facebook, Instagram and a community of individual developers and corporations. It's used for handling the view layer for Web and mobile apps. ReactJS allows us to create reusable UI components. END

References

[1] *http://www.hongkiat.com/blog/latest-webdev-tools/*
[2] *http://db-engines.com/en/blog_post/55*
[3] *https://angular.io/*
[4] *https://nodejs.org/en/*
[5] *http://db-engines.com/en/ranking/search+engine*

By: Roopendra Vishwakarma

The author currently works at Cognizant Technology Solutions as a projects associate. He has written many articles on various technologies, open source software, etc. He can be reached at *roopendra@techieroop.com*

Continued from page 71...

Embedded C

The standard known as Embedded C is slightly different from all the others. C from K&R C to C11 depicts the changes of a programming language over time, based on user requirements. But the Embedded C standard was proposed to customise the C language in such a way that it can cater to the needs of embedded system programmers. While the other standards of C are improvements over previous standards, Embedded C is a standard that is being developed in parallel. A lot of non-standard features were used in C programs written for embedded systems. Finally, in 2008, the C Standards committee came up with a standardisation document that everyone has to adhere to. Embedded C mostly has the syntax and semantics of normal C with additional features like fixed-point arithmetic, named address spaces, and basic I/O hardware addressing. Two groups of fixed-point data types called the *fract* types and the *accum* types were added to the C language to support fixed-point arithmetic. The keywords *_Fract* and *_Accum* are used for this purpose. New header files like <*iohw.h*> have also been proposed by the Embedded C standards committee. The main advantage of the Embedded C standard is that it is simpler and easier to learn than traditional C.

If you want to learn more about any of these standards, just search the Internet, particularly the website *http://www.open-std.org/*. I will be very happy if this article motivates somebody to migrate from ANSI C to C11, which was released in 2011. Six years have already elapsed, so it is high time to adopt the C11 standard. As an aside, the latest standard of C++, informally known as C++17, will be finalised in 2017. So, we have the chance to be the early adopters of a new standard of a very popular programming language, rather than be one among the late majority, as in the past.

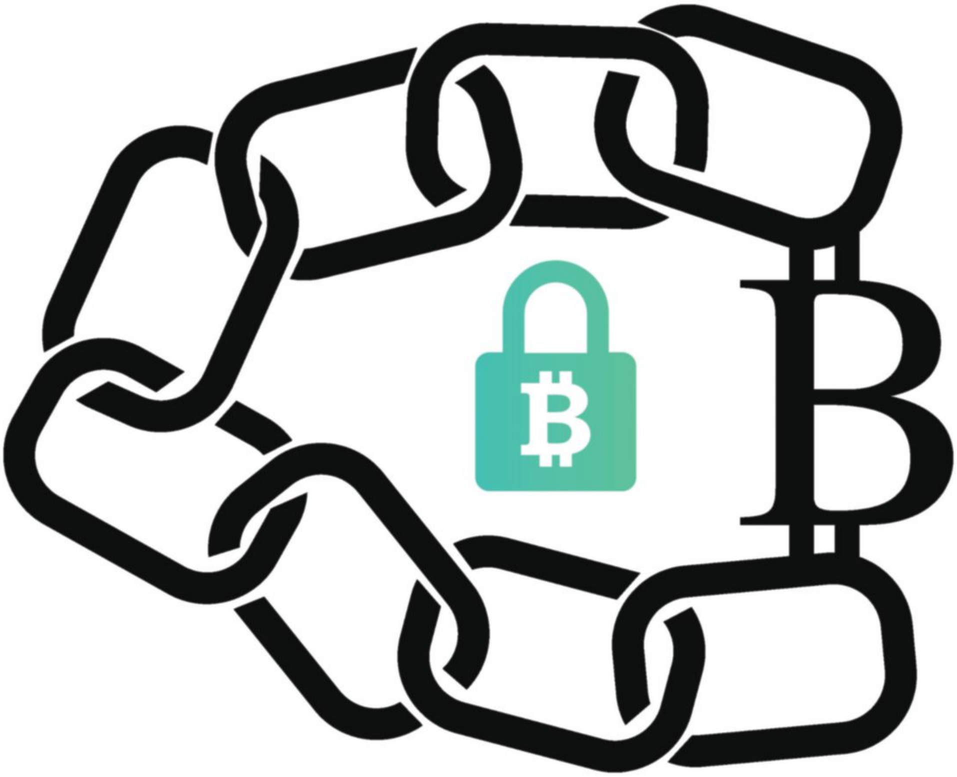
Before winding up, I want to mention the Turbo C compiler from Borland, which is outdated yet used by a lot of people. Which is the standard supported by the Turbo C compiler? Well, just like for the default standard supported by *gcc*, the answer is also 'None'. The default standard is the whole of ANSI C and some Turbo C specific extensions. Remember, these extensions are not available with *gcc* and this is the reason why header files like <*conio.h*> are not available in Linux. Finally, it took me some time to come up with this pun, "Read this article carefully and you will see that you are not all at sea about C!" END

By: Deepu Benson

The author currently works as an assistant professor in Amal Jyothi College of Engineering, Kanjirappally. He maintains a technical blog at *www.computingforbeginners.blogspot.in* and can be reached at *deepumb@hotmail.com.*

Blockchain-IoT based Management of Shared Resources between Trustless Parties

In this article, the authors discuss use cases for which blockchains and IoT devices can be used, instead of intermediaries, for Trustless parties to transact with each other. The authors also discuss the architecture and the open source technologies that were used in their solution.



Governments, communities or companies often need to track, control and manage assets and resources that they have a shared interest in. For example, in the logistics industry, a perishable item might get transported from a sender to a receiver. While in transit, the perishable item may have to be kept under a certain temperature or pressure. The item may get transported through multiple modes such as ship, train, truck, etc, and multiple transport operators might be involved before it reaches its destination. Payments have to be released to these parties if and only if they transported the item as per the contract.

Often, there is no trust between the sender, receiver and the various parties involved. This means that the parameters required to preserve a perishable item have to be tracked and controlled. And there needs to be a verifiable record of this tracking/controlling throughout the transportation period. There is also a need for a trusted way to settle payments between the various parties once the shipment has been delivered as per the contract.

Another use case is in the energy sector, where smart grids are used for the distribution and consumption of electric power. There is a need for power generators and distributors to track the supply and distribution of power, and keep records of it, to eventually reconcile and settle accounts between the producers and distributors of power. A third use case could be to manage water resources stored in dams and shared between countries/states.

The list of such use cases is endless. Often these parties do not trust each other and end up appointing an intermediary third-party trusted by all of them to manage, control or track the resources, on their behalf. The intermediaries, however, may charge exorbitant fees for their services or may become

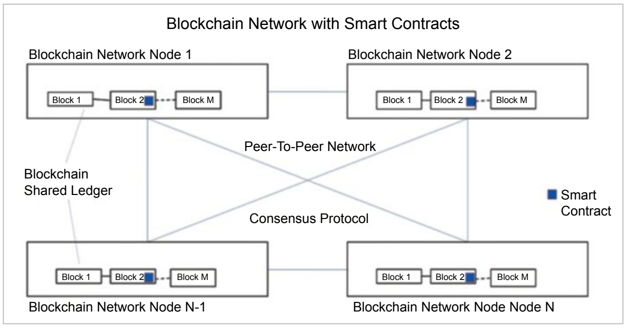

Blockchain Network with Smart Contracts
Blockchain Network Node 1
Blockchain Network Node 2
Block 1
Block 2
Block M
Peer-To-Peer Network
Blockchain
Shared Ledger
Smart
Contract
Consensus Protocol
Blockchain Network Node N-1
Blockchain Network Node Node N

Figure 1: Blockchain with smart contract

ineffective due to political/commercial reasons. Blockchain and IoT technologies together provide a solution which can do away with the intermediaries and autonomously track, control and manage resources in a trusted way.

Blockchain

The blockchain is a peer-to-peer network. It is based on a shared ledger (the blockchain database) and a protocol to keep the shared ledger updated. It is the technology underlying Bitcoin, the famous crypto-currency. Bitcoin was created by an unknown person nicknamed Satoshi Nakamoto. For people to accept and adopt Bitcoin as a crypto-currency, they had to be able to trust it. But, there is no government backing this crypto-currency. So, the trust has to be provided by the Bitcoin network itself.

The technology that provides such trust for Bitcoin is the blockchain. It is a shared ledger which is available on each full node of the network. In a Blockchain, each block is analogous to a ledger page and the blockchain is analogous to a ledger book. When a Bitcoin transaction happens on the network, the miner nodes verify the validity of the transaction, arrive at a consensus and update their copy of the ledger. The transactions verified and entered in the blockchain are immutable. Since blockchain is a peer-to-peer network, it does not have a central server. It has only peer level client applications, which join the blockchain network as nodes.

A blockchain provides some key benefits, which can be leveraged to provide solutions for use cases such as the ones discussed earlier in this article. Since it is a shared ledger on a peer-to-peer network, it does not have any central server which can be shut down unilaterally by any one party or authority. The records, transactions and smart contracts updated in the blockchain are immutable. They cannot be modified. The shared ledger is updated based on a consensus protocol and hence it prevents users or those transacting with Bitcoins from making fraudulent entries in the blockchain. This provides the trust needed when parties unknown to each other have to transact without an intermediary. Here, the blockchain network provides the trust.

There are a few open source platforms such as Ethereum, Hyperledger and Stellar, using which we can create a blockchain network as a public network, whereby any one can join the network. Alternatively, it can also be a private network behind the firewall of an organisation so that only those within the organisation can join it; or as a permissioned blockchain, where only those who are permitted to join the blockchain can do so.

Smart contracts

The features of Blockchain, such as decentralisation, immutability, verification, consensus and not being able to shut it down, provide a trustable network which can replace any intermediary. Soon, many people realised that the blockchain has the potential to be used for applications beyond crypto-currency such as Bitcoin. It can be used as a platform to execute or enforce contracts between unknown and hence untrusted parties in the form of smart contracts. But this needed a Turing Complete blockchain platform so that complex business logic could be written as code. The Bitcoin blockchain is not Turing Complete by design to avoid hacks. Hence, smart contracts cannot be written on the Bitcoin blockchain. This limitation gave rise to other blockchain platforms which can be used to write smart contracts, such as Ethereum, Hyperledger, etc.

Smart contracts are posted on the blockchain network in the same way we send crypto-currency. The posted contracts are added to the shared ledger (blockchain database). Smart contracts implement complex business logic as code. Each smart contract will have an address to which messages can be posted. A smart contract executes upon getting the messages posted to its address.

A client application connected to the blockchain network can receive messages from real-world assets such as IoT devices and post them to the smart contract. On getting the message, the smart contract executes and sends the result back to the client application in the form of an asynchronous event. This event can be used to control/manage the real-world assets.

In a way, smart contracts are analogous to the standing instructions given to banks to perform tasks such as transferring money to another account on specific dates, paying utility bills, etc. But, smart contracts implement more complex business logic and leverage the properties of the blockchain such as decentralisation, immutability and trust.

IoT

IoT (Internet of Things) is a network of things in the physical world. These things may be devices which have sensors within them or attached to them. In the context of IoT, a 'thing' is a server which is capable of getting sensor data from the device and sending it to the backend applications. A 'thing' server can be anything – a car, a refrigerator, a machine, a surveillance camera, a fan, a light, etc. IoT uses a local gateway server to connect the various thing servers in a building, car, etc, to the backend application. Thing servers use various communication technologies such as RFID, NFC, Wi-Fi, Bluetooth, and

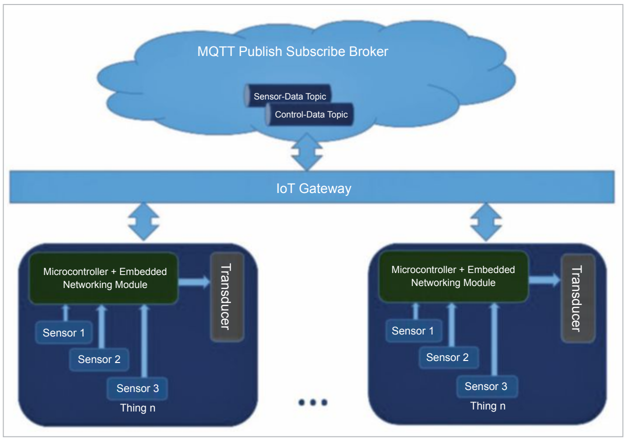


Figure 2: IoT thing server architecture

ZigBee to send data to the gateway server. Thing servers can also have wide area connectivity such as GSM, GPRS, 3G and LTE. On top of this communication infrastructure, IoT uses the MQTT protocol to connect the thing servers with the backend applications.

MQTT messaging protocol

The IoT thing servers may generate various types of data. Different backend applications might be interested in different types of data. The data exchange between the thing servers and the backend applications should happen asynchronously. There should not be any dependency on both of them to be alive at the same time to send and receive data.

To address this kind of a need, a publish/subscribe broker can be used in the architecture. HiveMQ is an open source publish/subscribe broker which uses the MQTT protocol to talk to the IoT thing servers on the one hand and to the backend applications on the other. The MQTT protocol works on the basis of topics. IoT thing servers can register topics with the HiveMQ broker. They will publish sensor data as messages on those topics. The backend servers will subscribe to those messages.

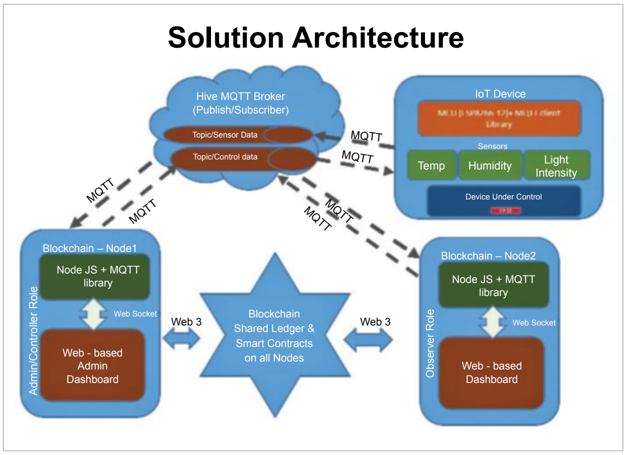


Figure 3: IoT-blockchain PoC architecture

Once an IoT thing server publishes a message on a specific topic, the HiveMQ broker sends that message to all the backend servers which have subscribed to that topic. Similarly, if the backend servers want to send any control information to the IoT thing servers, they can register the control topics with the HiveMQ broker and the IoT thing servers can subscribe to those topics. This will establish a two-way communication between the IoT things and the backend servers using the MQTT protocol and the HiveMQ broker.

PoC

We did a Proof of Concept (PoC) implementation using IoT, blockchain and MQTT to control a transducer. The building blocks of the PoC, and the open source technologies with which they were implemented, are explained below:

- A cloud based HiveMQ server was used as the MQTT publish/subscribe broker. On this broker, the IoT thing server registers a topic on which it publishes the sensor data and the blockchain node registers a topic on which it publishes the control data. In addition, the IoT thing server subscribes to receive the control data while the blockchain node subscribes to receive the sensor data. The HiveMQ broker delivers the published messages to subscribers.
- ESP8266-12 was used as the IoT thing server. This had a temperature, humidity and light sensor. It registered the sensor data topic with the cloud based HiveMQ MQTT broker and published the sensor data on this topic. It subscribed to the control messages from the blockchain node.
- A Node.js Web application, which subscribes to the sensor data topic on the HiveMQ broker for getting the sensor data, was used. It registers the topic on which it publishes the control messages on the HiveMQ broker. Subsequently, it publishes the control messages received from the blockchain. This Web application was connected to an Ethereum blockchain using the web3 JavaScript library. It registers the smart contract on the blockchain. It also posts the sensor data to the smart contract and watches for the asynchronous control messages from the smart contract. The control messages received from the smart contract are published through the HiveMQ broker.
- The smart contract was written using the Solidity language and compiled using a Solc compiler. The compiled smart contract is posted to the Ethereum blockchain using the web3 JavaScript library from the blockchain node.

Logistics use case

Let us again look at the logistics use case mentioned at the beginning of this article and see how the solution can be implemented. The perishable items are packed in a container in which a thing server with the temperature/pressure sensors is fitted. The sensor data is published through the MQTT broker.

Continued on page 81...

It's Easy to Build an App with Ember.JS!

This article presents a tutorial on how to build a simple app using the Ember.JS framework.

Ember.js is a popular, free and open source JavaScript Web framework, based on the model-view-view-model (MVVM) pattern. Although Ember.js is primarily considered a framework for the Web, it can also be used to build mobile and desktop applications. With over 17,000 stars and over 3400 forks at the time of writing this article, it is a very popular framework on GitHub. Moreover, it has some great features, which we will explore further in this article.

Ember.js is one of the front-end stack components built by the Ember core team. Here are some great features of the EmberJS framework.

Ember CLI

The Ember CLI (command line interface) allows the user to generate a new Ember app with the default stack. This utility provides:

- A standard directory and file structure.
- A server with live reload, which means it will automatically rebuild and reload apps whenever the files are changed.
- Support for ES6 modules.
- ES6/ES7 syntax support via Babel.
- Ember CLI testing framework.
- Dependencies managed via npm and Bower.
- Blueprints, which is a code generator for creating models and controller — components that are needed in an app.
- More features are available when using Ember CLI add-ons, of which there are over 2,000 available.

Ember Data

- Ember Data is a data-persistence library providing facilities for an object relational mapping to the Ember app.

Ember Inspector

This is a Web browser extension, available for Mozilla Firefox and Google Chrome. It makes debugging Ember applications easier, enabling users to watch template changes (in which views and components are currently rendered), and see the properties of Ember objects along with a UI, which also allows users to access the app's objects in the console itself.

Fastboot

Fastboot is an Ember CLI extension, which allows users to run their Ember apps in Node.js. Currently, this is in the alpha stage — once available, this feature will allow users to render the UI much faster.

Liquid Fire

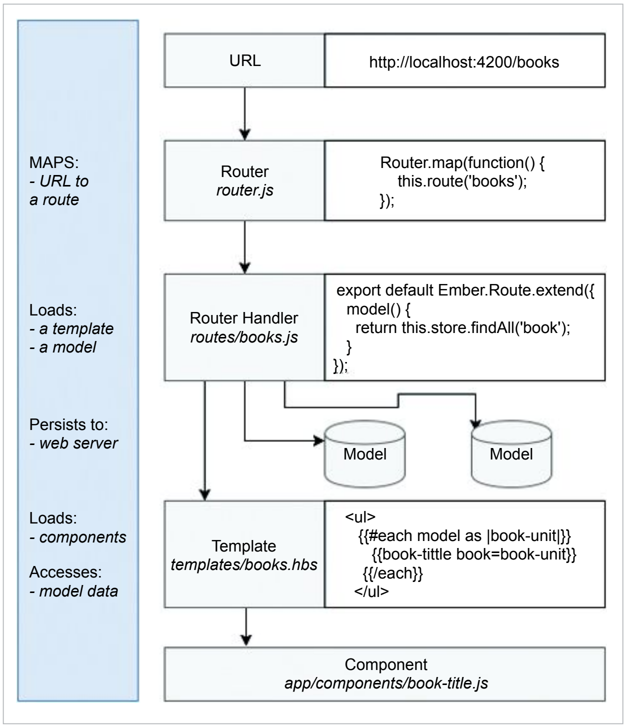
Liquid Fire is a toolkit which allows animated transitions in an Ember app.

Note: It is assumed that you have some basic knowledge of Web technologies like HTML, CSS and JavaScript. If you don't, W3Schools (*http://www.w3schools.com/*) is a good place to start. The site has some great tutorials for Web technologies that are easy to follow.

Before we start developing our sample app, let us have a look at the core concepts of the EmberJS framework, which are shown in Figure 1.

Route and route handlers

In Ember, the state of an app is represented by a URL. When the Ember app starts, the router is responsible for displaying templates, loading data or setting up the application state. It



```
Router.map(function() {
  this.route('books');
});
```

```
export default Ember.Route.extend({
  model() {
    return this.store.findAll('book');
  }
});
```

```
<ul>
  {{#each model as |book-unit|}}
    {{book-tittle book=book-unit}}
  {{/each}}
</ul>
```

Figure 1: The core concepts of the Ember.JS framework

does this by matching the current URL to the routes we've defined. Each URL has a corresponding route object that controls what users can see.

The Ember router maps the URL to a route handler, which renders a template and also loads a model, which is then available to the template.

Templates

Like the AngularJS app, the Ember app also uses templates for organising the layout of HTML. The templates in an Ember app look like any HTML fragment.

An example is:

```
<div>This is a valid template </div>
```

Ember templates use the syntax of Handlebar templates. The Handlebars syntax is used to build the DOM app. Templates can also display properties provided to them by the context.

Models

Models represent a persistent state. Every route has a model associated with it, which contains the data associated with it along with the current state of the app. Models can be configured to be saved somewhere else, like in the browser's local storage.

Components

Components control how the user interface works. Components consist of two parts—a template layout written in the Handlebars syntax, and a source file written in JavaScript that defines the logic behind the controller.

Installing Node.js

To use Ember CLI to build our app, Node.js must be installed first. Download and install Node.js (*https://www.nodejs.org*). The Ember CLI is distributed as an npm package, so we will use Node and npm to install the Ember CLI.

Installing Ember

Open a terminal/command prompt and type the following command:

```
c:\>npm install -g ember-cli
```

This command will install EmberJS and all its dependencies.

Testing the installation

After installing the Ember CLI via npm, let's check if everything is installed properly. Once we have installed the Ember CLI, we will have access to the 'ember' command in the terminal/command prompt. Let's use the 'ember new' command to create a new sample application by running the following command:

```
c:\>ember new my-project
```

This will create a directory called 'my-project', and will set up a default new Ember application inside it. This application will create default configuration files and will also include the following:

- Development server
- Template compilation
- CSS and JavaScript minification
- The ES2015 feature via Babel

Now let us check if everything is working fine. Change the working directory to the 'my-project' directory by running the following command:

```
c:\>cd my-project
```

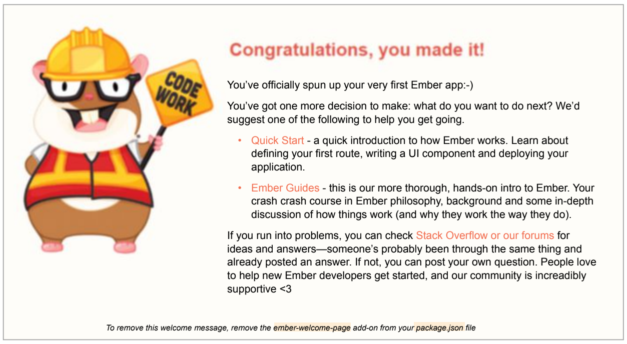
Now, start the development server by running the following command:

```
c:\my-project>ember server
```

After a few seconds, you will see the output in your terminal or command prompt, and it will look like what's shown below:

```
Livereload server on http://localhost:49152
Serving on http://localhost:4200/
```

Open *http://localhost:4200* in your browser. You should see an Ember welcome page, as shown in Figure 2.



Congratulations, you made it!

You've officially spun up your very first Ember app:-)

You've got one more decision to make: what do you want to do next? We'd suggest one of the following to help you get going.

- Quick Start - a quick introduction to how Ember works. Learn about defining your first route, writing a UI component and deploying your application.
- Ember Guides - this is our more thorough, hands-on intro to Ember. Your crash crash course in Ember philosophy, background and some in-depth discussion of how things work (and why they work the way they do).

If you run into problems, you can check Stack Overflow or our forums for ideas and answers—someone's probably been through the same thing and already posted an answer. If not, you can post your own question. People love to help new Ember developers get started, and our community is increadibly supportive <3

To remove this welcome message, remove the ember-welcome-page add-on from your package.json file

Figure 2: The default welcome page of the EmberJS framework

Getting started with a sample app

To start building a sample app, create a new template in the 'my-project' directory, using the following command in a terminal or at the command prompt:

```
C:\my-project>ember generate template application
```

This command creates a template called 'application' at *app\templates\application.hbs* in the 'my-project' directory. This template is always loaded with your app and is always on screen.

Now open *app\templates\application.hbs* in your favourite editor and add the following code:

```
<h1> Open Source For You Magazines </h1>
{{outlet}}
```

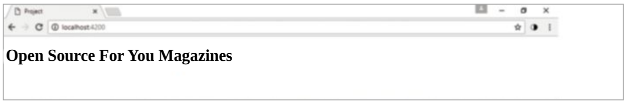
This is the main template or, in simple words, 'main page' of our sample app. When we visit our sample app via the URL we will see this template first. We have also added '{{outlet}}' to this template to render a route in that place.

Ember CLI has a watcher, which automatically detects and reloads the page in the background. When you run the server using the following command, you will see that the welcome page is replaced by *'Open Source For You Magazines'*.

```
C:\my-project>ember server
```

After a few seconds, you will see the output in your terminal or command prompt, which looks like what's shown below:

```
Livereload server on http://localhost:49152
Serving on http://localhost:4200/
```



Open Source For You Magazines

Figure 3: Default welcome page replaced by our app's page

Open *http://localhost:4200* in your browser. You should see a new page, as shown in Figure 3.

Defining a route

Let us build a simple app, which shows a list of issues for 'Open Source For You Magazine'. For this we need to create a route first. In simple words, routes are just different pages that make up your application.

To generate a route, run the following command in the terminal or command prompt:

```
C:\ project>ember generate route magazines
```

After running this command, you'll see an output like what's shown below:

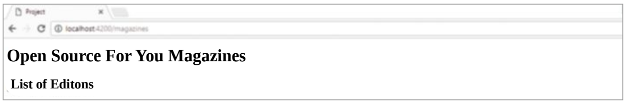
```
installing route
  create app\routes\magazines.js
  create app\templates\magazines.hbs
updating router
  add route magazines
installing route-test
  create tests\unit\routes\magazines-test.js
```

This means Ember CLI has created a template for 'magazines', which will be displayed when the user visits *http://localhost:4200/magazines*. It also adds a unit test for this route.

Now open the created template in *app/templates/* called *magazines.hbs* and add the following code:

```
<h2>List of Editions </h2>
```

Open your browser, start the server and go to *http://localhost:4200/magazines*. You should see the rendered content of *magazines.hbs* along with our main application template, as shown in Figure 4.



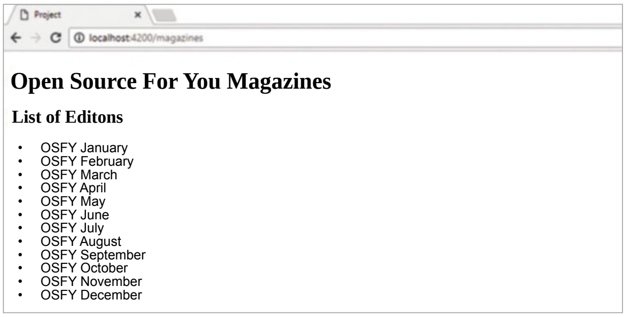
Open Source For You Magazines

List of Editons

Figure 4: Displaying rendered content of the route template

Now that we have rendered the magazine's template, let's give it some data to render. We do that by specifying a model for that route, and by editing *app/routes/magazines.js*. Copy the following code into *magazines.js*:

```
import Ember from 'ember';

export default Ember.Route.extend({
  model() {
    return ['OSFY January', 'OSFY February', 'OSFY March',
     'OSFY April', 'OSFY May', 'OSFY June', 'OSFY July',
     'OSFY August', 'OSFY September', 'OSFY October',
```



Open Source For You Magazines

List of Editons

- OSFY January
- OSFY February
- OSFY March
- OSFY April
- OSFY May
- OSFY June
- OSFY July
- OSFY August
- OSFY September
- OSFY October
- OSFY November
- OSFY December

Figure 5: Displaying rendered content along with data in a template

```
    'OSFY November', 'OSFY December'];
  }
});
```

In the route's *model()* function, return the data you want to make available for the template. In this case, pass the list of monthly *OSFY* issues to the template.

Now, we will render the array of strings, returned by *model()* method, into HTML. Open *magazines.hbs* and add the following code:

```
<h2>List of Editions</h2>
<ul>
  {{#each model as |magazine|}}
    <li>{{magazine}}</li>
  {{/each}}
</ul>
```

We have used the Handlebar syntax to loop through the data and print it. We have also used each helper to loop over every item in the array we provided from the *model()* hook and print it inside an *<li>* element.

Now, open your browser, start the server and go to *http://localhost:4200/magazines*. You should see the rendered content of *magazines.hbs* along with our main application template, as shown in Figure 5.

References

[1] *http://emberjs.com/*
[2] *https://guides.emberjs.com/v2.10.0/*
[3] *https://en.wikipedia.org/wiki/Ember.js*

By: Aniket Eknath Kudale

The author has more than two years of experience as a software engineer at Tibco Software Inc., Pune. His interests include Web technologies, computer vision and security. You can reach him at *kudale@aniket.co*.

Continued from page 77...

The various transport operators together could form a consortium and start a blockchain network. In this case, the blockchain is a permissioned one, where the consortium controls who can join its blockchain. Only members of the consortium may be allowed to join the blockchain network. Alternatively, the public Ethereum blockchain network can be used and the smart contract can be deployed on it.

The transport operator may run a Web application on his node with an easy-to-use user interface to create the smart contract on the blockchain. The sender gets into a contract with the transport operator. The terms and conditions of the transportation and payment are coded as a smart contract and deployed on the blockchain. The payment may be locked up in an escrow account.

The sensor data from the shipped container is received by the blockchain nodes and posted to the smart contract, which verifies the recorded temperature/pressure parameters as per the codified terms of the contract. Upon successful delivery of the item, the smart contract will trigger the payment from the escrow account. The payment will be completed in near real-time.

In future, micro payments between machines and M2M (machine-to-machine) communication without human intervention will find wide application. Today's centralised client-server world is being augmented with decentralised, peer-to-peer, disintermediated digital solutions. Blockchains with smart contracts and IoT are the evolving technologies which will drive such an exciting world.

Acknowledgements

The authors would like to acknowledge and thank Sriram Rajagopalan and Sarav Daman of Wipro Limited for their critical review of the article and contribution during the implementation of the PoC.

References

[1] *https://www.ethereum.org/*
[2] *https://www.hyperledger.org/*
[3] *https://www.stellar.org/*
[4] *http://mqtt.org/*
[5] *http://www.hivemq.com/demos/websocket-client5.*
[6] *https://en.wikipedia.org/wiki/Internet_of_things*
[7] *http://solidity.readthedocs.io/en/develop/*
[8] *https://www.npmjs.com/package/solc*
[9] *https://slock.it/*

By: Venkatachalam Subramanian and Sumanta Basu

Venkatachalam Subramanian is a principal consultant in talent transformation, Wipro Limited, Bengaluru. He has 20 years of experience in the IT industry.

Sumanta Basu is a senior architect in the communications vertical, Wipro Limited, Bengaluru. He has more than 12 years of experience in the IT industry.

CONQUER THE WEB THE *REST* WAY

This article gives readers a basic introduction on how to use the REST API.

Today, the buzz word is ‘data’ and organisations owning large volumes of useful data are considered wealthy. Earlier, the data that a company owned was solely used within the organisation; but times have changed. The world now interacts and communicates through data — in the form of JSON, XML, Video, Excel and many more such formats. Many Twitter and Facebook APIs are consumed by developers every day for their own purposes. In Google Translate, the user can render a sentence in any language desired. A Twitter API can easily find the top trending tweets. To add to the list, there are several sites that give information about weather conditions merely by the user providing the geo location.

The handshake between the producer and the consumer of the data or the server and the client can be done in many ways but the most popular is by using RESTful Web services. A Web service is what is offered by one electronic device to another, communicating with each other via the World Wide Web. Almost every organisation today uses the RESTful Web services to provide its data as a service. In this article, we will create a simple Web service in Java and access it through Postman (the software to test the Web services).

The best way to start is to know the basics of REST Web services. REST is an architecture based Web service, which stands for Representational State Transfer and it uses different URIs to expose business logic. REST uses the HTTP methods like *Get* for read-only access to resources, *Put* to create new resources, *Delete* to remove the resource, *Post* to update or create a new resource, and *Options* to get supported operations on a resource. In REST, all the contents like text files, videos, images and HTML pages are treated as resources that are provided by the server. The REST client accesses and modifies them. REST uses different resource representations like JSON, XML and text.

In designing a RESTful Web application, it is pertinent to ask the question, “How RESTful is the API?” To answer this, we have a model developed by Leonard Richardson called the Richardson Maturity Model. It states that APIs can be categorised in four levels — 0, 1, 2 and 3. Level 0 or *Swamp of POX* states that there is one URI; the request body contains all the details and uses plain old XML. Level 1 is the starting point for REST and it has individual URIs for each resource. If we introduce different HTTP methods to do different operations on the resource URI, then it is called Level 2. Finally, if one implements HATEOS, i.e., the responses have links to further information that the client can use, the API is said to be at Level 3.

To develop a REST API we need JAX-RS (which is a bunch of interfaces and annotations implemented by different libraries like Jersey), RESTEasy or Apache Wink. You can use any library to develop the API but I will use Jersey.

Let’s set up the environment to kickstart the Web service. The prerequisites for the project are:

a) Java 1.6 and above
b) Eclipse EE version (Kepler)
c) A Web server (Tomcat, in our case)
d) Postman (a Chrome extension)

Start with downloading the latest version of Java and get it installed on your machine.

1. Java 1.8 can be downloaded from *http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html.*
2. Eclipse can be downloaded from *http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/keplerr.*
3. Create a Maven project in Eclipse named ‘REST’. We will create a Maven project as we want all the JARs to be downloaded and linked from the Maven repository. You can very well download the JARs and attach them to the project. To create the Maven project, select the archetype as Jersey (if you don’t find the Jersey archetype, then this is not registered in your development machine). You can very well add the archetype with the details shown in Figure 2. After completing the Maven project, download the Apache

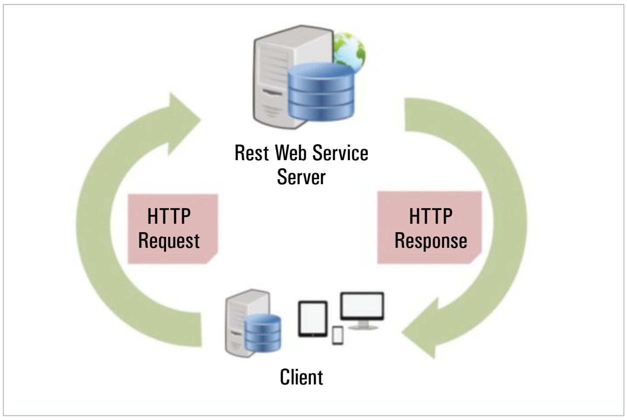
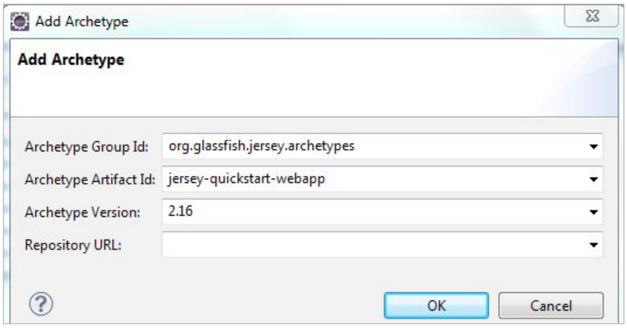


Figure 1: REST



Add Archetype

Add Archetype

Archetype Group Id: org.glassfish.jersey.archetypes

Archetype Artifact Id: jersey-quickstart-webapp

Archetype Version: 2.16

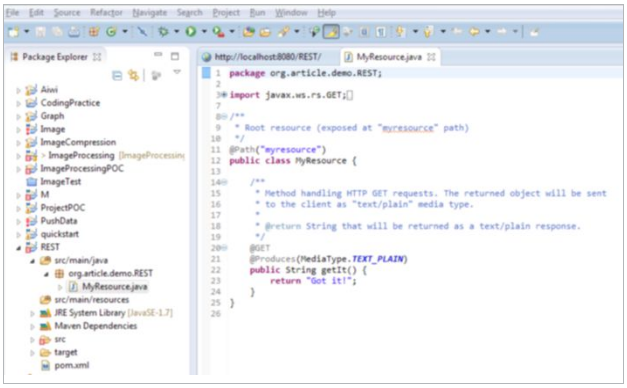
Repository URL:

Figure 2: Archetype

Tomcat server, where the REST services will be deployed. Click on the link *https://tomcat.apache.org/download-70.cgi* to download it. In my case, it was 64-bit Windows Zip. Install the Tomcat server, integrate it to Eclipse and then start it.

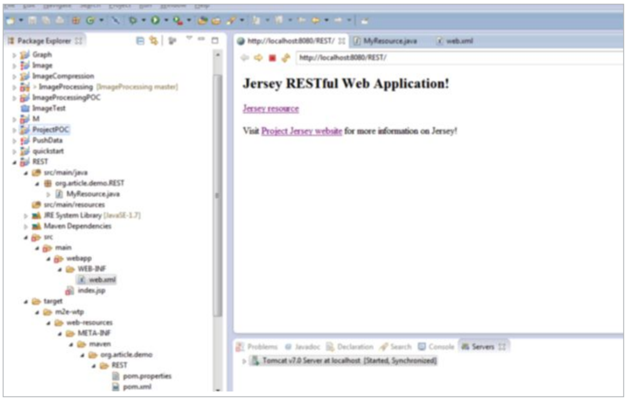
Install Postman, which comes as a Chrome plugin. Click on *Settings* in the Chrome browser and go to *Extensions*. Scroll down and click *Get more extensions*. Search for *Postman* in the *Search* tab and add it to Chrome.

Now, let's start coding. In the Maven project, you will find *MyResource.java* that has the *getIt()* method which returns 'Got it!' by default, as shown in Figure 3. This Java file has three annotations named *@Path*, *@Get* and *@ Produces*. The *@Path* annotation identifies the URI path template to which the resource responds, the @Get annotation is a request method designator and corresponds to the HTTP method, and the *@Produces* annotation is used to describe the output which can be text, JSON or XML. Right-click on the project and run it on the server. This will display a page in the browser with the link *http://localhost:8080/REST/*, as shown in Figure 4. Upon clicking the *Jersey Resource* button you will be redirected to *http://localhost:8080/REST/webapi/ myresource* where you will see 'Got It!' as output. The reason behind this is that *web.xml* has the servlet mapping as */webapi*. The above URL can also be accessed through Postman to verify the output. In Postman, you can paste the URL, select the request type as *Get* and click the *Send* button. This will return 'Got It!' as the output.



```java
package org.article.demo.REST;

import javax.ws.rs.GET;

/**
 * Root resource (exposed at "myresource" path)
 */
@Path("myresource")
public class MyResource {

    /**
     * Method handling HTTP GET requests. The returned object will be sent
     * to the client as "text/plain" media type.
     *
     * @return String that will be returned as a text/plain response.
     */
    @GET
    @Produces(MediaType.TEXT_PLAIN)
    public String getIt() {
        return "Got it!";
    }
}
```

Figure 3: Default resource



http://localhost:8080/REST/

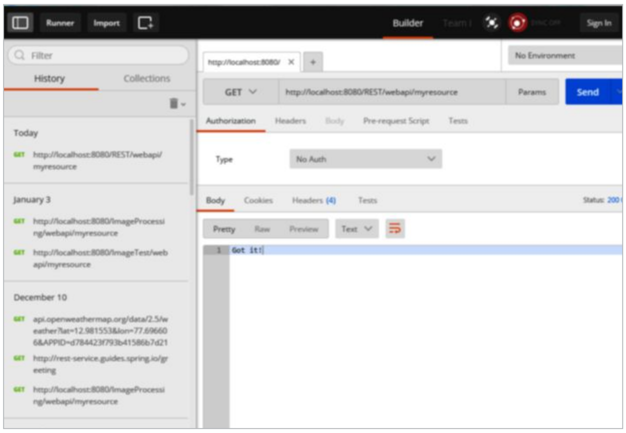
Jersey RESTful Web Application!

Jersey resource

Visit Project Jersey website for more information on Jersey!

Tomcat v7.0 Server at localhost [Started, Synchronized]

Figure 4: Default Web Page



GET http://localhost:8080/REST/webapi/myresource

Type: No Auth

Status: 200

```
Got it!
```

Runner
Import
Builder
Sign In
Filter
History
Collections
No Environment
Params
Send
Authorization
Headers
Body
Pre-request Script
Tests
Body
Cookies
Headers (4)
Tests
Pretty
Raw
Preview
Text
Today
http://localhost:8080/REST/webapi/myresource
January 3
http://localhost:8080/ImageProcessing/webapi/myresource
http://localhost:8080/ImageTest/webapi/myresource
December 10
api.openweathermap.org/data/2.5/weather?lat=12.981553&lon=77.696606&APPID=d784423f793b41586b7d21
http://rest-service.guides.spring.io/greeting
http://localhost:8080/ImageProcessing/webapi/myresource

Figure 5: Postman

We can conclude that in order to access the resources from different portals, we need Web services, and one of the most widely used is REST Web services. Its security features, like user authentication and header checks, are of particular importance. END

By: Ashish Kumar Sinha

The author is a software engineer based in Bengaluru. A software enthusiast by heart, he is passionate about using open source technology and sharing it with the world. He can be reached at *ashi.sinha.87@gmail.com*. Twitter handle: *@sinha_tweet*

Creating a Digital Wallet Application in App Inventor 2

This month, we present another interesting app in our App Inventor 2 series. Digital wallets are a trending topic ever since our country decided to opt for a 'less cash' economy. Creating this app will give readers a good insight into how a digital wallet really works.

After reading and practically trying out the apps in this series, many readers have mastered the components available in the palette and have had much fun moving them between the designer and block editor.

Digital wallets or digital payment services are the talk of the town nowadays because of the government's initiatives to go digital and the Digital India campaign. Vendors are opting for mobile payments rather than cash, and this is a smooth as well as hassle-free process. Let's explore the functionality of digital wallet payment systems with the help of a mobile application which we will develop over the course of this two-part tutorial.

Before that, I would like to define what a digital wallet is and how it works. A digital wallet is an application or feature within an electronic device which enables it to make electronic transactions. It has a linked bank account number from which all the transactions are processed and each transaction is recorded for future reference. The device needs to be connected to the Internet to make payments.

The theme of the application

The theme is pretty simple and you might have already come across various mobile payment applications. We will make a similar app, and demonstrate its debit and credit features in the digital wallet account.

GUI requirements

For every app, we need a graphical user interface or GUI, which helps the user to interact with the on-screen components. How each component responds to user actions is defined in the block editor section. In this app, we will have multiple screens and will define the look and feel for each of them.

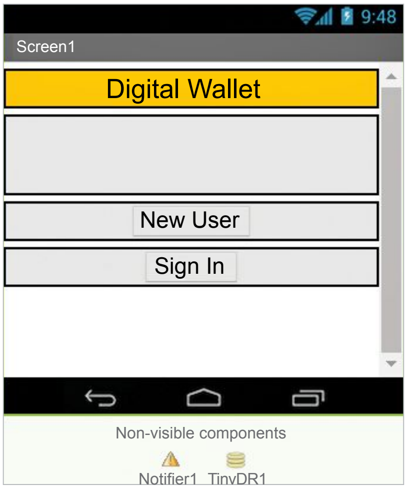


Figure 1: Designer screen (Screen1)

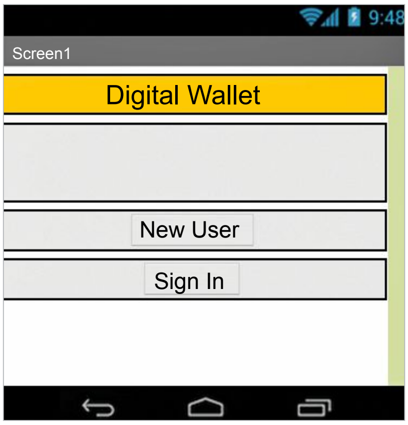


Figure 2: How the application looks

There are four GUI screens. Their names are:

1. Screen1
2. Login Screen
3. New User Screen
4. Purchase Screen

The GUI requirements for the application are listed below.

1. *Label:* Labels are static text components, which are used to display some headings or markings on to the screen.
2. *Button:* Buttons will let you trigger the event and are very essential components.
3. *Horizontal arrangement:* These are special components which keep all child components aligned within themselves. The horizontal component keeps all the child components horizontally aligned.
4. *Notifier:* These are used to display some instructions or to give controls over existing components. You will discover more about their functionality as we implement them in our application.
5. *TinyDB:* This is the internal storage within the device, which will be used to keep confidential data such as passwords, etc.
6. *Text Boxes:* These are pretty simple

Table 1

	Component name	Purpose	Location
1	Label	To display a label	Palette-->User Interface-->Label
2	Button	To trigger events	Palette-->User Interface-->Button
3	Horizontal arrangement	To arrange the child components	Palette-->Layout-->Horizontal Arrangement
4	Notifier	To display on-screen information	Palette-->User Interface-->Notifier
5	TinyDB	To store data	Palette--> Storage--> TinyDB

Table 2

	Component name	Purpose	Location
1	Label	To display a label	Palette-->User Interface-->Label
2	Button	To trigger events	Palette-->User Interface-->Button
3	Horizontal arrangement	To arrange the child components	Palette-->Layout-->Horizontal Arrangement
4	Notifier	To display on-screen information	Palette-->User Interface-->Notifier
5	Text box	To take user inputs	Palette-->User Interface-->Text Box
6	TinyDB	To store data	Palette--> Storage--> TinyDB

objects that are used to take inputs from the user. You can better control the type of text box by setting various properties such as numeric-only or multiple lines.

Table 1 lists the components that we will require for Screen1, which we will drag on to the designer from the left hand side palette.

1. Drag and drop the components mentioned in Table 1 to the viewer.
2. Visible components will be visible to you while the non-visible components will be located beneath the viewer under the tag 'Non-visible'.
3. We have placed a label so as to put the name of the application.
4. All buttons need to be put within the *Horizontal arrangement* so as to keep them aligned horizontally.
5. If you have dragged and placed everything, the Screen1 layout will look somewhat like what's shown in Figure 1.
6. Make the necessary property changes for the label and button components so that they best fit on to the screen.

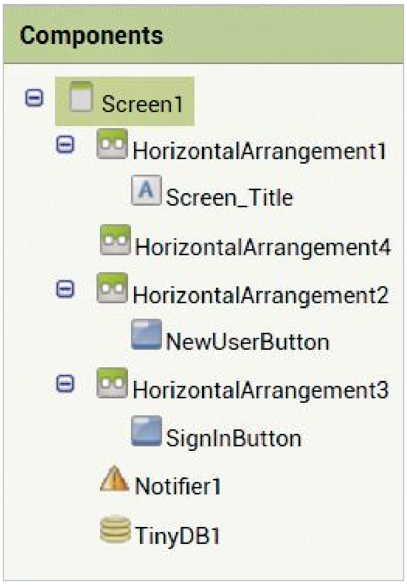


Figure 3: Components view (Screen1)

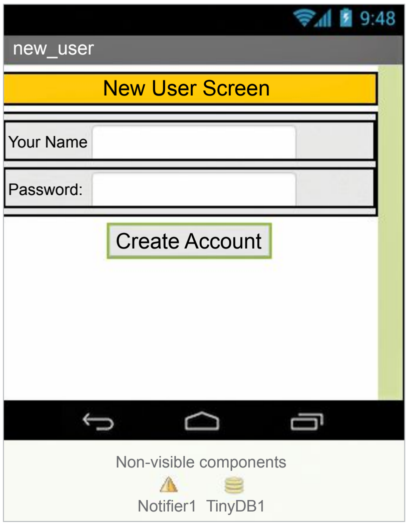


Figure 4: Designer screen (New user screen)

7. Renaming the components helps to identify them in the block editor.
8. So, this way, your graphical user interface is ready for the first screen. Figure 2 shows exactly how the application will look after the installation.
9. The hierarchy of the components that we have dragged to the designer is as shown in Figure 3.

In a similar fashion, we will design the remaining three screens. Table 2 lists the components that we will require for the new user screen, which we will drag on to the designer from the left hand side palette.

If you have dragged and placed everything, the new user screen layout will look something like what's shown in Figure 4.

Given in Figure 5 is the hierarchy of the components that we have dragged to the designer.

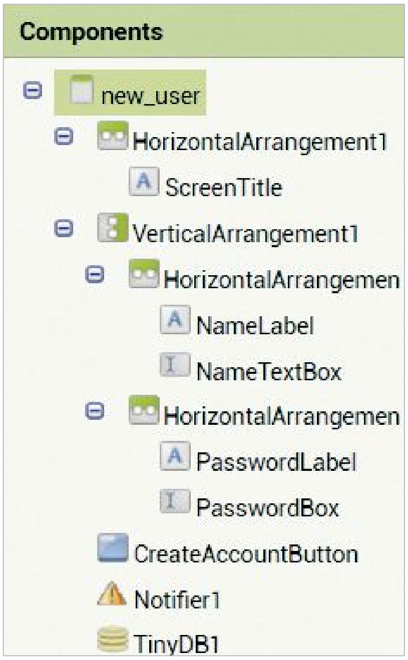


Figure 5: Components view (New user screen)

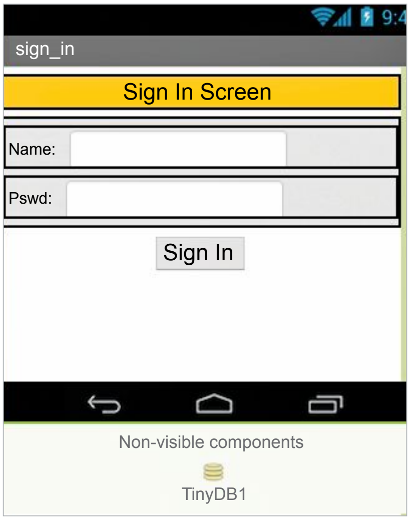


Figure 6: Designer screen (sign-in screen)

The components that we will require for the sign-in screen which we will drag on to the designer from the left hand side palette are given in Table 3.

Table 3

	Component name	Purpose	Location
1	Label	To display a label	Palette-->User Interface-->Label
2	Button	To trigger events	Palette-->User Interface-->Button
3	Horizontal arrangement	To arrange the child components	Palette-->Layout-->Horizontal Arrangement
4	Notifier	To display on-screen information	Palette-->User Interface-->Notifier
5	Text box	To take user inputs	Palette-->User Interface-->Text Box
6	TinyDB	To store data	Palette--> Storage--> TinyDB

Table 4

	Component name	Purpose	Location
1	Label	To display a label	Palette-->User Interface-->Label
2	Button	To trigger events	Palette-->User Interface-->Button
3	Horizontal arrangement	To arrange the child components	Palette-->Layout-->Horizontal Arrangement
4	Notifier	To display on-screen information	Palette-->User Interface-->Notifier
5	List picker	To select an option from a list	Palette-->User Interface-->ListPicker

If you have dragged and placed everything, the sign-in screen layout will look something like what's shown in Figure 6.

Figure 7 shows the hierarchy of the components that we have dragged to the designer.

The components that we will require for the purchase screen, which we drag on to the designer from the left hand side palette, are given in Table 4.

If you have dragged and placed everything, the purchase screen layout will be similar to Figure 8.

The hierarchy of the components that we have dragged to the designer is given in Figure 9.

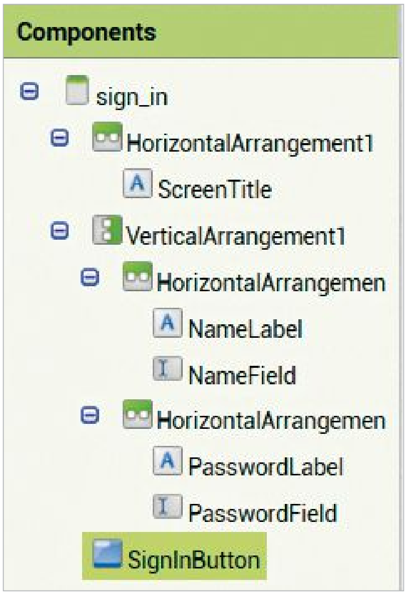


Figure 7: Components view (sign-in screen)

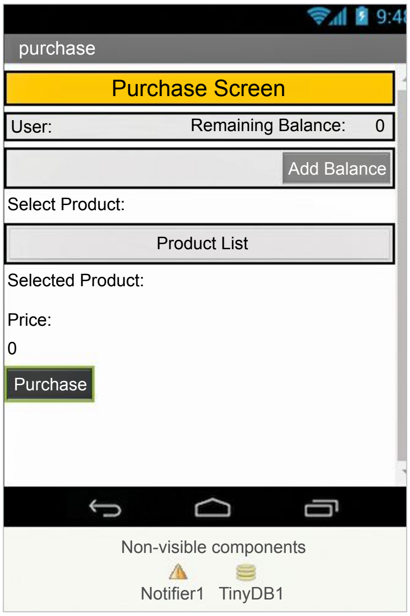


Figure 8: Designer screen (purchase)

So we have been able to develop a simple GUI, which will be enough to demonstrate how a digital transaction app works. Next month, we will explore the logic to be implemented for each of the screens.

If readers have uploaded any of their applications to Google Play Store, I would be pleased if you mailed me the links.

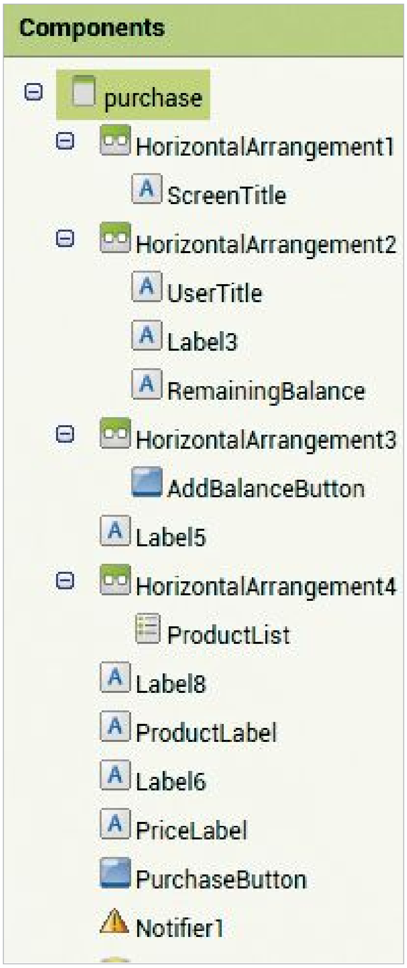


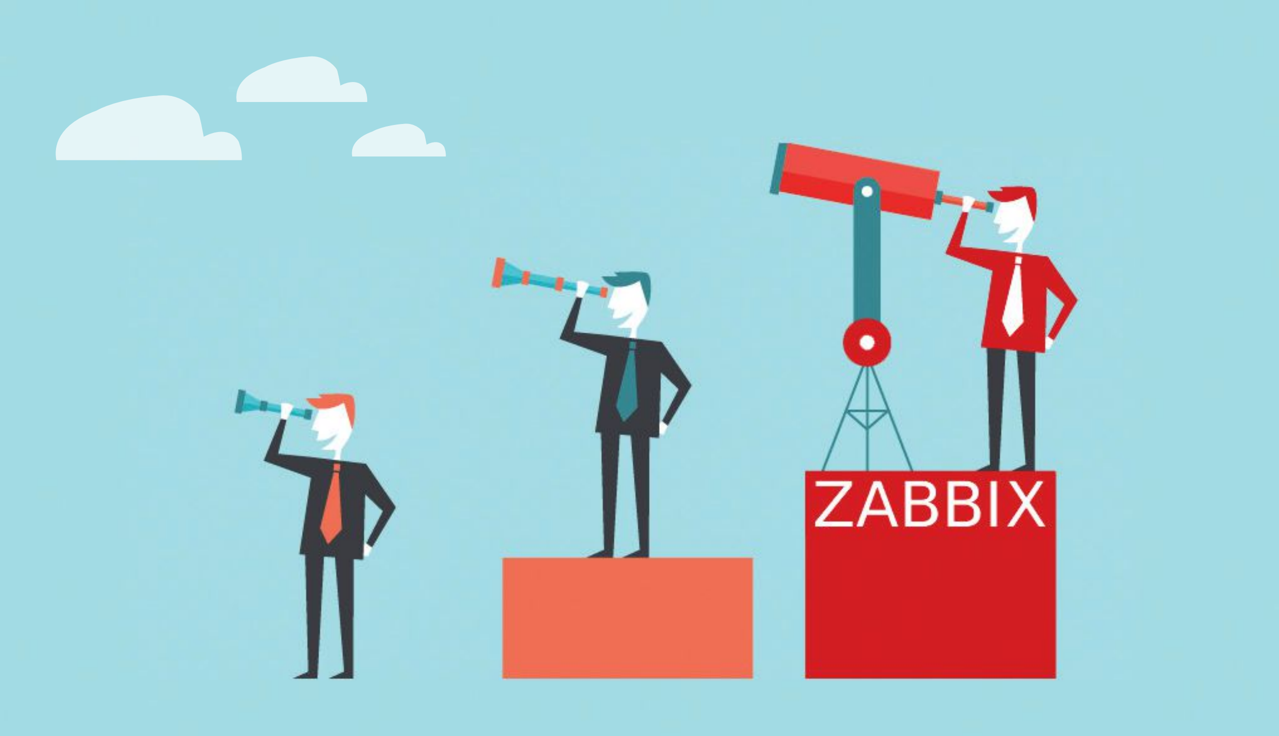
Figure 9: Components view (purchase)

By: Meghraj Singh Beniwal

The author is a B. Tech in electronics and communication, a freelance writer and an Android app developer. He is currently working as an automation engineer at Infosys, Pune. He can be contacted at *meghrajsingh01@rediffmail.com* or *meghrajwithandroid@gmail.com*

An Introduction to ZABBIX

Here's an introduction to Zabbix, the open source monitoring tool. Zabbix is a highly integrated network monitoring solution, which has an array of features in a single package.



Zabbix is open source network software that provides agents to monitor remote hosts and includes support for monitoring via SNMP, TCP and ICMP checks. It offers real-time monitoring of thousands of metrics collected from servers, virtual machines and any other kind of networking device. Its capability ranges from monitoring the traffic in the network to tracking how much ink is left in your printer. It also offers excellent reporting and data visualisation features based on the stored data.

Zabbix was created by Alexei Vladishev and is currently being actively developed and supported by Zabbix SIA.

An overview of Zabbix

Zabbix uses the client-server architecture and a small agent on the monitored client to gather data and send it to the Zabbix server. Zabbix version 3 supports encrypted communication between the server and connected clients, so that data is protected while it travels over insecure networks.

Zabbix consists of several major software components. These components and their features are outlined below.

Server: The Zabbix server is the central component to which agents report availability and integrity information and statistics. The server is the central repository in which all configuration, statistical and operational data is stored.

Database storage: All configuration information as well as the data gathered by Zabbix is stored in a database.

Web interface: For easy access to Zabbix from anywhere and from any platform, a Web based interface is provided. The interface is part of the Zabbix server and usually (but not necessarily) runs on the same physical machine as the one running the server.

Proxy: The Zabbix proxy can collect performance and availability data on behalf of the Zabbix server. A proxy is an optional part of the Zabbix deployment; however, deploying it may be very beneficial to distribute the load of a single Zabbix server.

Agent: Zabbix agents are deployed on monitoring targets to actively track local resources and applications, and report the gathered data to the Zabbix server.

Zabbix features

Zabbix is a highly integrated network monitoring solution, with an array of features in a single package. Listed below are some of its features:

1. Data gathering
2. Real-time graphing
3. Web monitoring
4. Network discovery

5. Audit logging
6. Easy configuration
7. Agent-less monitoring
8. Web interface
9. Extensive visualisation options
10. JMX monitoring

Configuring Zabbix

There are primarily four ways of getting Zabbix on your system:

1. Installing it from distribution packages.
2. Downloading the latest source archive and compiling it yourself.
3. Installing it from the containers.
4. Downloading the virtual appliance.

Requirements

Memory: Zabbix requires both physical and disk memory. A minimum of 128MB of physical memory and 256MB of disk memory are required to start it.

CPU: Zabbix, especially the database, may require significant CPU resources depending on the number of monitored parameters and the chosen database engine.

Zabbix can easily be run on a number of operating systems like:

- Linux
- IBM AIX
- FreeBSD
- NetBSD
- OpenBSD
- Mac OS
- Solaris
- Windows

Zabbix version releases

The first public version of Zabbix was released in 2001 and was called Zabbix 1.0alpha1. But the first stable version 1.0 was released in 2004. After this, a new stable release came out every one-and-a-half years. The latest Zabbix 3.2.3 version was released on December 21, 2016. The release date of various versions can be obtained from the Zabbix website. END

By: Neetesh Mehrotra

The author is employed at TCS as a systems engineer. He is interested in Java development and automation testing. He can be contacted at *mehrotra.neetesh@gmail.com.*

Exploring Front-end Computer Vision

Computer vision tasks include methods for acquiring, processing, analysing and understanding digital images, and in general, deal with the extraction of multi-dimensional data from the real world in order to produce numerical or symbolic information.

Computer vision (CV) is a discipline that relates to the visual perception and analysis of computers and cameras. The visual input method for computers is a camera. A majority of all computer vision algorithms focus on extrapolating interesting features from images/videos that are captured by a camera. This field has many applications in the field of robotics. For example, the preliminary versions of Stanford University's Stanley (a self-driving car) used a pair of stereo cameras for visual perception.

Technology today is shifting to a more cloud and Internet oriented setting. Traditional software is being replaced by Web apps. If eventually, everything is going to be ported to a Web platform, it would be wise to start incorporating the Web into upcoming technologies. Similarly, one could think of shifting CV to a browser platform as well. In fact, there are various libraries that provide browser based support for computer vision. These include *Tracking.js*. First, let it be clear that a browser based system for this article refers to front-end code only, involving just HTML5, CSS, and JavaScript.

Basic computer vision and the browser

Computations are carried out upon images, with the fundamental unit being a pixel. Algorithms involve mathematical operations on a pixel or a group of pixels. This article addresses a few hackneyed CV algorithms and their ports to a front-end system. To start with, basic concepts like images and canvas are to be understood first.

An HTML image element refers to the '<img></img>' tag. It is, essentially, adding an image to a Web page. Similarly, to process or display any graphical units, the '<canvas></canvas>' element is used. Each of these elements has attributes such as height, width, etc, and is referred to via an ID. The computation part is done using JavaScript (JS). A JS file can be included either at the head or body of an HTML document. It contains functions that will implement the aforementioned operations. For drawing any content upon a canvas, a 2D rendering reference called context is supposed to be made.

Here's how to access images, as well as canvas and context, from JS:

```
//getting image, canvas and context
var im = document.getElementById("image_id");
var canvas = document.getElementById("canvas_id");
var context = canvas.getContext("2d");

//accessing a rectangular set of pixels through context
interface
var pixel = context.getImageData(x, y, width, height);

//displaying image data
context.putImageData(image, start_point_x, start_point_y);
```

Using a local Web cam from a browser

Accessing a Web cam from the browser first requires user consent. Local files with a URL pattern such as *file:///* are not allowed. Regular *https://* URLs are permitted to access media.

Whenever this feature is executed, the user's consent will be required. Any image or video captured by a camera is essentially media. Hence, there has to be a media object to set up, initialise and handle any data received by the Web cam. This ability of seamless integration is due to media APIs provided by the browser.

To access the Web cam with a media API, use this code:

```
navigator.getUserMedia = (
           navigator.getUserMedia ||
           navigator.webkitGetUserMedia ||
           navigator.mozGetUserMedia ||
           navigator.msGetUserMedia );
```

In the above code, *navigator.getUserMedia* will be set if the media exists. To get control of media (refers to camera), use the following code:

```
navigator.getUserMedia({
     video: true
},handle_video, report_error );
```

On the successful reception of a frame, the *handle_video* handler is called. In case of any error, *report_error* is called.

To display a frame, use the following code:

```
var video_frame = document.getElementById("myVideo");
video_frame.src = window.URL.createObjectURL(stream); //
stream is a default parameter
```

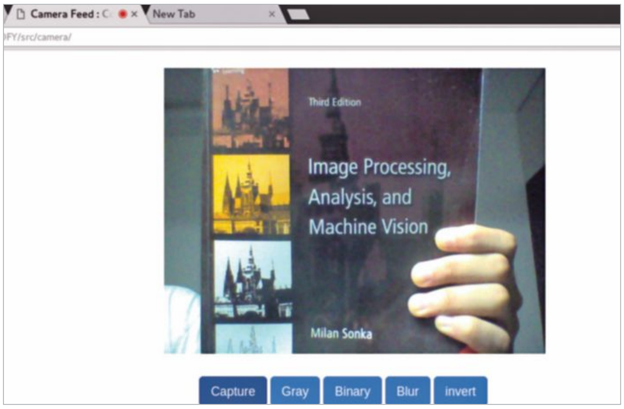


Figure 1: Displaying a frame

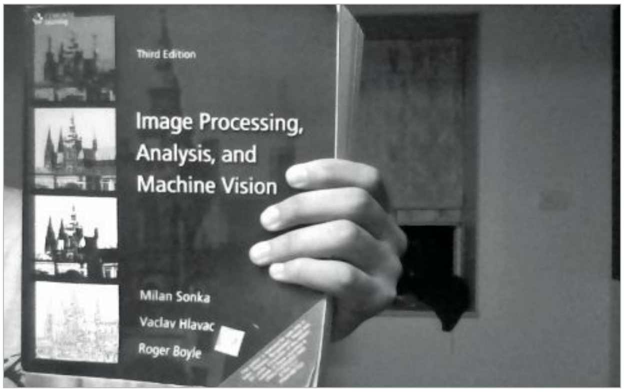


Figure 2: Grayscale image

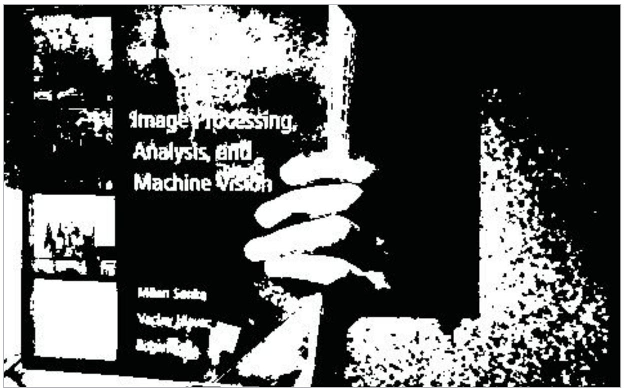


Figure 3: Binary image

```
//provided to handle_video
```

For further details, regarding a camera interfacing with the browser, refer to *https://github.com/kushalvyas/trackingjs_ofy/*.

The basic image processing algorithms

JS stores an image as a linear array in RGBA format. Each image can be split into its respective channels, as shown below:

```
var image = context.getImageData(0, 0, canvas.width, canvas.
height);
var channels == image.data.length/4;
```

```
for(var i=0;i<channels;i++){
     var red_component_pixel = image.data[i*4 + 0];
     var green_component_pixel = image.data[i*4 + 1];
     var blue_component_pixel = image.data[i*4 + 2];
}
```

Computation of gray scale images

A gray scale image is one in which all colour components are normalised to have equal weightage. If an 8-bit image is considered, the colour gray is obtained when the number of RGB bits equals 1.

To solve this, there is a simple formula, which creates a weighted sum of pixel values to yield a gray image:

```
gray[pixel] = 0.21*red_component_pixel + 0.72*green_
component_pixel +    0.07*blue_component_pixel'
```

On applying the above formula to each pixel, split into its components, one gets an equivalent gray pixel.

Computation of binary and inverted images

A binary image is in black and white (BW). The conversion of an image from colour to BW is done through a process called thresholding, which classifies each pixel as white or black based on its value. If the value is greater than a particular threshold, it will be set to 255, else 0.

```
if(red_component_pixel > threshold_red &&
     green_component_pixel > threshold_green &&
       blue_component_pixel > threshold_blue){
     //make pixel == white
     image.data[pixel] = 255;
}else{ image.data[pixel] = 0; }
```

Just as we have negatives for a photograph, similarly, the inversion of colour space of any image converts all pixels into a negative. This can simply be done by subtracting each pixel value from 255.

The *tracking.js* library

According to GitHub, *tracking.js* is a lightweight JS library that offers a variety of computer vision algorithms with HTML5 and JS. Some algorithms implemented here are for colour tracking, face detection, feature descriptors and the other utility functions. To set up *tracking.js* for your Web page, include *build/tracking.js* inside your '<head>'. For more details, one can visit *tracking.js* documentation. It is highly detailed and illustrated.

Colour tracker using *tracking.js*: To initialise a colour tracker, first use the following commands:

```
var myTracker = new tracking.ColorTracker(['yellow']);
myTracker.on("track", color_tracking_callback);
var mT = tracking.track("#myVideo", myTracker);
```

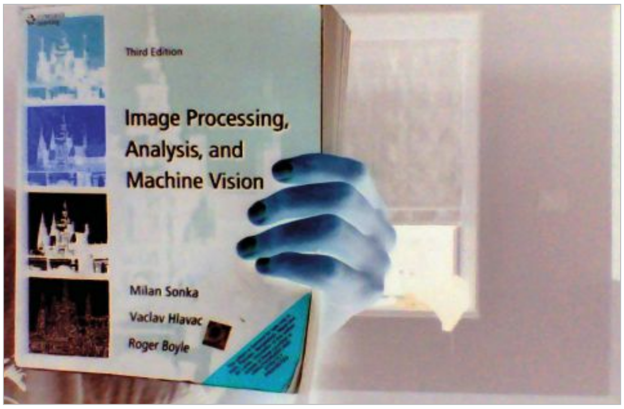


Figure 4: Inversion

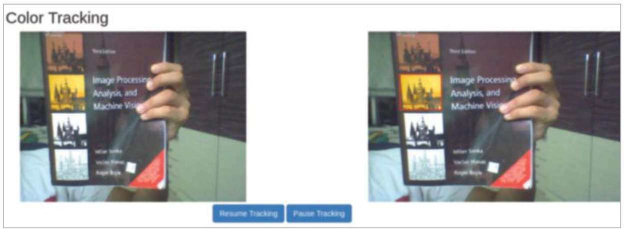


Figure 5: Yellow colour based tracking of the book

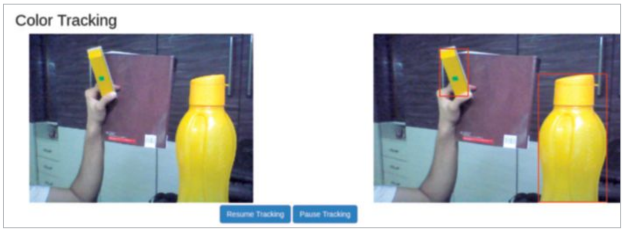


Figure 6: Multiple colour region tracking

In the above code snippet, *color_tracking_callback* is a callback which will receive a list of all possible locations where the given colour is present. Each location is a rectangle object, comprising attributes which are 'x, y, width and height'. x and y are the starting points of the rectangle.

The natural action for tracking is to make a bounding box around the region we are interested in. Therefore, the *boundingBox* function plots a rectangle around the region of interest. Context variable is used here to perform any canvas drawing methods. *context.stroke()* eventually prints it on the canvas.

```
function color_tracking_callback(list_rect){
     list_rect.data.forEach(drawBoundingBox);
}

function drawBoundingBox(rect){
     context.beginPath();
     context.strokeStyle = "red";
     context.lineWidth = "2";
     context.rect(rect.x, rect.y, rect.width, rect.height);
```

```
    context.stroke();
}
```

Starting and pausing the tracking process

To start the tracking process, *tracking.js* provides a call to *start( )* and *stop( )* methods.

```
mT.stop(); //to stop tracking
mT.start(); //to start tracking
```

Setting up custom colours for tracking

As seen, the input to a colour tracker is a list of probable colours (e.g., [*yellow*]). As the definition suggests, a colour tracker must be able to track colours. *Tracking.js* provides a method *registerColor* that handles user-specified custom colours.

```
tracking.ColorTracker.registerColor('<color_name>' ,
callback_color);
```

The =*callback_color* callback will have input arguments as red, blue and green values. Since this is a custom colour, one has to define the RGB ranges. If the RGB argument meets the range, the function returns true, else it'll return false.

```
function callback_color(r , g, b){
    if(r > r_low && r < r_high && g > g_low && g < g_high &&
b > b_low && b < b_high){
      return true;
    }
    return false;
}
```

Here, *r_low, r_high*, etc, refer to the lower and upper bounds of the threshold values, respectively. Having registered the colour, one can simply append *color_name* to *color_list* in *tracking.ColorTracker (color_list).*

Face tagging using *tracking.js*

Facebook has this feature whereby one can tag one's friends. There are different sets of mathematical frameworks developed to perform visual recognition as well as detection, of which one of the most robust options is the Viola-Jones Detection framework.

A brief introduction to Viola Jones: Each human face has multiple features, with many significant discernible visual differences which are the inputs that help in face recognition. These are known as Haar Cascades (which you can look up in */src/detection/training/haar*). Examples for significant variations in facial features include:

- Location of the eyes and nose
- Size of the eyes, nose, etc
- Mutual contrast between facial features

By training over such features, the detection framework is made to locate areas of an image containing regions that satisfy the above constraints, thereby aiding in face detection.

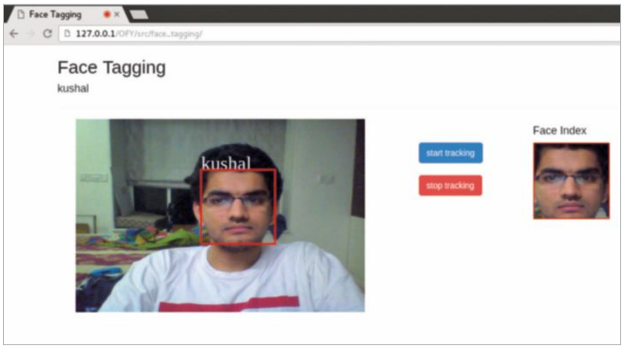


Figure 7: Face detection, tagging and tracking

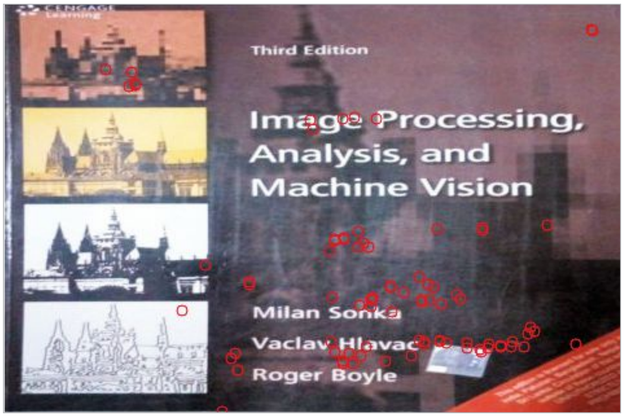
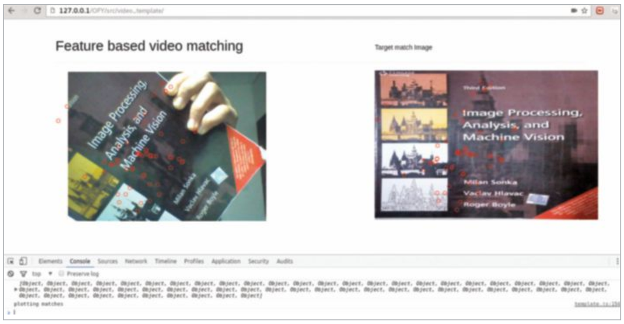


Figure 8: Feature points



```
[Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object,
Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object, Object,
Object, Object, Object, Object, Object, Object, Object, Object, Object, Object]
plotting matches
```

Figure 9: Matching via features

To integrate your front-end with face recognition, *tracking.js* provides another script located in *build/data/face-min.js*. This basically loads the Viola Jones parameters over trained data, including *face-min.js* as well as *tracking.min.js* files.

To initialise and use the face tracker, type:

```
var face_tracker = new tracking.ObjectTracker("face");
face_tracker.setInitialScale(param_for_block_scaling);
face_tracker.setEdgesDensity(classifier_param_for_edges_
inside_block);
face_tracker.setStepSize(block_step_size);
var mTracker = tracking.track("#myVideo", face_tracker,
{camera:'true'});
face_tracker.on("track", handle_faces);
```

The function *handle_faces* is a callback fired for handling detected regions. As mentioned earlier, *tracking.js* returns a list containing Rect objects. In the application discussed, the detected faces will be tagged via a JavaScript prompt. Once the prompt value is taken, the face is identified and tracked with the given name as well as indexed for UI purposes. The complete code can be obtained at *//githublink*. If the face is detected initially, or there is a state change of tracking (stop/start), the prompt is re-called and the data is stored within an array. For tracking purposes, each newly obtained Rect object is compared with the previously recorded nearset face. Comparison is based on the minimum Euclidean distance. If not returned, then it is recalculated.

Features extraction and matching

In simple terms, any significant discernible parts of the image can be defined as a feature. These can be corner points, edges or even a group of vectors oriented independently. The process of extracting such information is called feature extraction. Various implementations exist for feature extraction and descriptors, such as SIFT, SURF (feature descriptors) and FAST (corner detection). *Tracking.js* implements BRIEF (Binary Robust Independent Elementary Features) and FAST (Features from Accelerated Segmentation Test). Input to the system is first a gray image. The following code extracts corner points (points of interest) based on FAST.

```
var gray = tracking.Image.grayscale(input_image, width,
height);
var corners = tracking.Fast.findCorners(gray, width, height);
```

Each feature point can be referred to as a location. But to be able to perform any operations, these locations are converted into descriptors, which can be considered as a list of vectors that define a given feature. Comparison operators are applied upon these vectors. To find descriptors, *tracking.js* uses the BRIEF framework to extrapolate descriptor vectors from given feature points.

```
var descriptors = tracking.Brief.getDescriptors(gray, width,
corners);
```

Having got the points of interest from an image as well as their descriptors, we can design a scenario wherein one can track based on templates. Given a video frame and a fixed image, features can be used to match and identify where the fixed image can be located. However, there can be false positives.

```
var matches = tracking.Brief.reciprocalMatch(corner_scene,
descriptor_scene ,corner_target, descriptor_target);
// calculates the matching points between the scene and the
target image.
```

```
matches.sort(function(a, b){
     //matches can be further filtered by using a sorting
functin
     // Either sort according to number of matches found:
     return b.length – a.length;
     // or sort according to confidence value:
     return b.confidence – a.confidence
}
```

The matches obtained can be sorted on the basis of their length, i.e., the number of matches obtained, and on their confidence value, as to how well the points match. Having arranged the matches, efficient matching of the target template image and the scene image can be carried out. It is simply a task of graphics now. Just iterate over the two arrays and mark the appropriate feature points on the canvas, as follows:

```
function plot_matches(matches){
     for (var i = 0; i < matches.length; i++) {
                    var color = "red";
                   context.lineWidth = "2px";
                   context.fillStyle = color;
                   context.strokeStyle = color;
              context.beginPath();
              context.arc(matches[i].keypoint1[0], matches[i].
keypoint1[1], 4, 0, 2*Math.PI);
        context.stroke();
     }
}
```

The above function plots the matches only for the scene image, since the reference context is made with respect to one canvas element. For plotting matches on the target template image, a context reference has to be made to its respective canvas element.

Computer vision on the front-end can be used for various applications, not only to produce various image effects but also applications like browser based gesture control, etc. The advent of JavaScript libraries has helped to make this possible. The code can be found at *https://www.github.com/kushalvyas/trackingjs_ofy* END

References

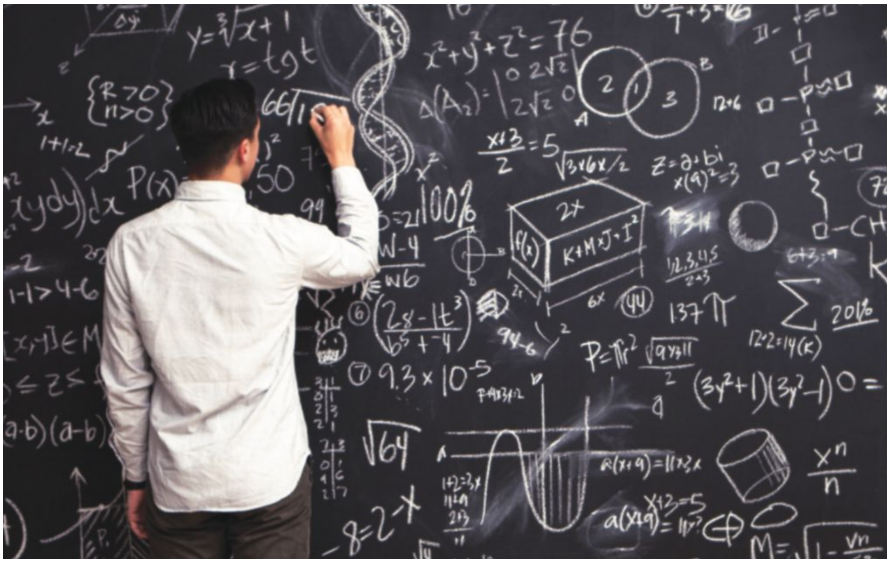
[1] Sebastian Thrun. Udacity lecture on artificial intelligence.
[2] *www.trackingjs.com/*
[3] *https://developer.mozilla.org/en-US/docs/Web/API/Navigator/getUserMedia*
[4] *https://github.com/eduardolundgren/tracking.js*
[5] *https://github.com/kushalvyas/trackingjs_ofy/*

By: Kushal Vyas

The author is an open source enthusiast. His projects involve image searching and recognition, visual tracking and 3D reconstruction. He has worked with many open source frameworks such as OpenCV, *tracking.js*, Boost libraries, Django, etc. He blogs at *www.bitsmakemecrazy.com*.

Lex: Waving the Magic Wand of Regular Expressions

Lex is a computer program that generates lexical analysers. It is commonly used with the Yacc parser. Originally distributed as proprietary software, some versions of Lex are now open source.



I would like to start by saying that the creator is as important as the creation. Lex and Yacc are the creators of a compiler and an interpreter, respectively. To be specific, Lex is used in the first phase of compilation, which is called the lexical analysis phase, as a lexical analyser generator; while Yacc is used in the second, parsing phase as a parser generator. You may feel that since you are not into compilers, this article is of no use for you. Yet, if you use the pattern matching algorithm, this article will help you.

The lexical analysis phase involves the processing of words, the core of which is the identification of patterns in the input. You might have used different pattern matching algorithms. When searching for a single pattern, if we use C for the search, a nested loop containing a conditional statement may be required, though that may not be so simple. Ten patterns may need a minimum of 10 conditional statements if a simple line of code is required for checking a pattern. This is where Lex comes in with the wand of regular expressions.

Lex was developed by Mike Lesk and Eric Schmidt in 1975. According to them, Lex is a program generator designed for lexical processing of character input streams. This is done with the help of regular expressions. Lex identifies the input strings based on the regular expressions specified in the program. Sample programs in this article were run using Flex. Depending on the version used, commands and programs may vary slightly.

Installing Lex

Installation in Linux: Lex is proprietary software, though open source alternatives to it are available. Flex or fast lexical analyser is one such package. To install Flex, type the following command in the terminal:

```
sudo apt-get install flex
```

Installation in Windows: In Windows, installation is done by simply downloading the package and running the program. When you install, make sure that the package is not installed in the folder *Program Files*.

Regular expressions

Before going into the details of Lex, a quick overview of regular expressions is required. The name itself implies that the regularity of something can be expressed using a regular expression. Let's suppose we have the following set of sentences:

- Sharon is 27 years old.
- Stephy is 29 years old.
- Sunny is 56 years old.
- Shyamala is 46 years old.

We can see that in all the above sentences, the first word begins with 'S' followed by a sequence of characters. The next word is 'is', then comes a two digit number which is followed by 'years old'. This pattern can be represented using a regular expression such as:

```
S[a-z]+ is [0-9]{2} years old.
```

Here, *S[a-z]+* denotes that one or more occurrences of any character from 'a' to 'z' can follow 'S', and *[0-9]{2}* represents any two-digit number. Note that blank spaces are used to separate each word in the sentences and are used similarly in the regular expression also.

According to Wikipedia, a regular expression is a sequence of characters that define a search pattern, mainly for use in pattern matching with strings, or string matching. The following are some of the notational short-hand symbols used in a regular expression:

- The '.' (dot) is the wild card character that matches any single character.
- | stands for 'or'. A|b means either A or b.
- - stands for a range of characters.
- '?' is used to represent zero or one instance. 'cars?' can represent either car or cars.
- + represents one or more instances of an entity.
- * represents zero or more instances of an entity.
- ( ) is for grouping a sequence of characters.
- Character classes: [abc] represents a| b| c.

Given below are the rules for naming an identifier in C::

- It must begin with an underscore or an alphabet.
- It can be followed by zero or more underscores, a letter of the alphabet or a number.

We can write the regular expression as:

```
[a-zA-Z_]([a-zA-Z_0-9])*
```

Intelligent people always find easy ways to get their job done. It may be difficult to repeatedly write 'a-zA-Z' for letters of the alphabet. Hence, the idea emerged to give names to regular expressions that are frequently used and are lengthy.

Regular definitions allow one to give names to regular expressions. These short names help to avoid repeatedly writing lengthy regular expressions.

As an example, l -> a-zA-Z and d -> 0-9. Given this regular definition, a regular expression for identifiers can be re-written as:

```
( l | _)( l | d |_)*
```

The structure of a Lex program

To quote from the book of Genesis, where the Biblical version of the creation of the world is described, "And God said, 'Let there be light' and there was light." Likewise, the Lex compiler said, "Let there be the file lex.yy.c: and there was lex.yy.c." But wait, before discussing lex.yy.c, we must discuss the structure of a Lex program, which consists of the following sections:

- Definition section
- Rule section
- User defined subroutines

The definition section: The definition section starts with *%{* and ends with *%}*. We can write C program segments within this section. Everything contained within this section will be directly copied to the generated C file. Conventions for comments in C can be followed for inserting comments in this section. If we write comments outside *%{ %}*, we must indent them with a single white space for the Lex compiler to understand them as comments.

```
e.g. %{
           /* This is a sample program.
         This will be copied to C file
         This section contains  # define, extern, and global
declarations (if any)

     */
     %}
```

This section may also contain regular definitions. But the regular definitions are defined outside *%{ %}*. It is optional to include regular definitions.

The rule section: This section starts and ends with *%%*. All the rules for translation are defined within this section. Each rule has a pattern and an action part separated by a white space. The patterns are specified using a regular expression. '|'(a vertical bar) can be used to separate patterns for which the action is the same.

For example:

```
abc |sdf {printf("strings");}
```

…results in printing 'strings' if the input contains either 'abc' or 'sdf'. The rules are considered in the order in which they are specified. Let me explain this point with the help of an example.

The input contains *"int a=10; char name[10]="sharon" "* and rules are specified as:

```
%%
[a-zA-Z]+ { printf("strings");}
int|char|break {printf("keywords");}
%%
```

The first rule says that any combination of letters from A to Z or a to z can be identified as a string. So 'a', 'name', and 'sharon' will be identified as strings. Even though 'int' and 'char' are keywords, they too are combinations of letters from a to z, resulting in Lex identifying them as strings. Now let us change the order of the rules:

```
%%
int|char|break {printf("keywords");}
[a-zA-Z]+ { printf("strings");}
%%
```

This set of rules will identify 'int' and 'char' in the input file as keywords and 'a', 'name', and 'sharon' as strings for the same input. Thus the order in which rules are specified does matter in Lex programs.

Another notable feature is that Lex patterns will match a given input character or string only once and execute the action for the longest possible match.

For example, the input contains a string 'intimacy'.

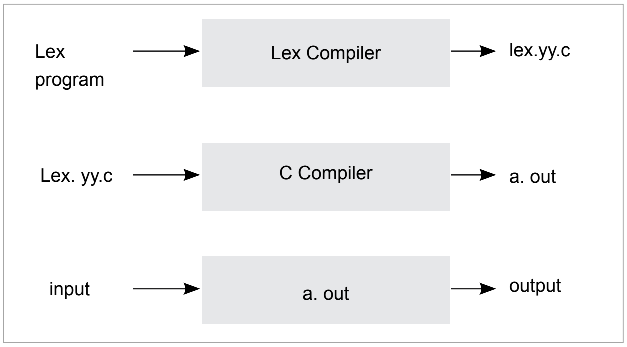


Figure 1: How the system works

It can be thought of as resulting in printing 'int' as the keyword and 'imacy' as the string. But since Lex carries out the action for the longest possible match, 'intimacy' will be identified as the string.

User defined subroutines: This section includes the definition of the *main( )* function and other user defined functions which will be copied to the end of the generated C file. The definition of *main()* is optional, since it is defined by the Lex compiler.

Program execution

A Lex file can be saved with the *.l* or *.lex* extension. Once the program is compiled using a Lex compiler, a C file *lex.yy.c* is generated. It is then compiled to get an *a.out* file, which accepts the input and produces the result. Diagramatic representation of the same is shown in Figure 1.

The following command is used to compile a program using the Lex compiler:

```
lex file_name.l
```

The C file generated -- *lex.yy.c* -- can be compiled using a gcc compiler:

```
gcc lex.yy.c -lfl
```

The above code generates the executable file *a.out*. The *lfl* or *ll* (depending on the alternatives to Lex being used) options are used to include the library functions provided by the Lex compiler.

```
./a.out < input_file_name
```

The above command will give the desired output.

Sample program 1

The following is a Lex program to count the number of characters in a file:

```
     %{
       int charCount;
     %}
//This comment is to tell that following is the rule section
     %%
     [a-zA-Z.] { charCount++;}
     %%
     main()
     {
       yylex();
       printf("%d  \n", charCount);
     }
Sample input_1:  abs. +as
Sample output_1: +7
Sample input_2:  abs.as
Sample output_2: 6
```

All the user defined variables must be declared. In this example, the variable charCount is declared in the definition section. Otherwise, when the *lex.yy.c* file is compiled, the C compiler will report the error 'charCount undeclared'. The regular expression [a-zA-Z.] represents any letter in the English alphabet or a . (dot). Whenever a letter or . (dot) is encountered, charCount is incremented by one. It is to be noted that when you want to use . (dot) literally outside [ ], not as a wild card character, put a \(backslash) before it.

When you write the rule section, make sure that no extra white space is used. Unnecessary white space will cause the Lex compiler to report an error.

In the *main()* function, *yylex()* is being called, which reads each character in the input file and is matched against the patterns specified in the rule section. If a match is found, the corresponding action will be carried out. If no match is found, that character will be written to the output file or output stream. In this program, the action to be performed when '+' is encountered is not specified in the rule section. So it is written directly to the output stream. Hence sample output_1.

The *lex.yy.c* file

The following is a description of the generated C file *lex.yy.c*. The Lex compiler will generate static arrays that serve the purpose of a finite automata, using the patterns specified in the rule section. A switch-case statement corresponding to the rule section can be seen in the file *lex.yy.c*. The following code segments are from *lex.yy.c*, obtained by compiling sample program 1.

```
switch ( yy_act )
     { /* beginning of action switch */
           case 0: /* must back up */
           /* undo the effects of YY_DO_BEFORE_ACTION */
           *yy_cp = (yy_hold_char);
           yy_cp = (yy_last_accepting_cpos);
           yy_current_state = (yy_last_accepting_state);
           goto yy_find_action;
case 1:
YY_RULE_SETUP
```

```
#line 7 "eg3.lex"
{ charCount++; }
     YY_BREAK
case 2:
YY_RULE_SETUP
#line 8 "eg3.lex"
ECHO;
     YY_BREAK
```

This is generated corresponding to the rule section in the sample program. We can see that case 0 is for input retraction. This is needed because it may be that only after reading the next character does the compiler understand that the character read is part of the next token. In that case, the file pointer needs to be retracted. Case 1 includes the action specified in Rule 1, to increment the value of the variable charCount. Case 2 is for simply printing the unrecognised symbol to the output stream. *ECHO*, seen in Case 2, is a macro name and its definition is as follows:

```
#define ECHO do { if (fwrite( yytext, yyleng, 1, yyout )) {} }
while (0)
```

ECHO will simply write the content of *yytext* to the output stream. The use of *yytext* is mentioned in the next section.

The definition of *YY_BREAK*, a macro name seen in Case 2 of the above switch case block, is as follows:

```
/* Code executed at the end of each rule. */
#ifndef YY_BREAK
#define YY_BREAK break;
#endif
```

Thus *YY_BREAK* is to break the flow of execution.

Built-in functions and variables

The Lex compiler provides us with a number of built-in variables and functions. Some of them are listed below.

- The Lex library contains the definition of *main()* function. Programmers can either simply include *main()* from the library or can define their own *main()*. The *main()* function defined in the library contains a call to the function *yylex( )*. The default definition of *main()* is as follows:

```
main(int ac, char **av)
 {
          return yylex( ) ;
 }
```

The variable *yytext* is a character array that contains a character/group of characters read from the input file for matching with any pattern specified in the rule section.

- The function *yywrap( )* returns 1 if EOF is encountered, otherwise 0 is returned. One can modify this function according to one's needs so that it may return 1 until all the files required are read.
- The variable *yylen* is the length of string in *yytext*.
- Some alternatives of Lex define the variable *yylineno*. It keeps track of the line number which is required for error reporting. *yylineno* is updated when the pattern for the new line (*n*) is matched.

Sample program 2

The following program will make it clear how users can define their own sub-routines to use in a Lex program.

```
%%
[0-9]+ {display(yytext);}
. {}
%%
display(char a[])
{
printf("%s\n",a);
}
```

The above program prints nothing but the combination of digits in the input to the output stream. We can see that the second pattern specified in the rule section is a simple . (dot) and the action part specified as '{}'. This means whenever this pattern matches, do nothing. You can see the variable *yytext*, which contains the current lexeme, is passed as the argument to the user defined function *display()*. Another point to be noted is that this program does not contain a *main()* function. The default definition of *main()* must be included while this program is compiled. Otherwise an error will be generated. As stated earlier, the definition of *main()* can be included by compiling *lex.yy.c* with the *lfl* or *ll* options.

Lex is a very powerful tool that can be used to manipulate text that involves patterns. It gains its power from regular expressions. Luckily, the Lex language is compatible with C, a very commonly used programming language known to even high school students and thus, many of the problems solved using C can be coded in Lex. This simple tool makes the job of programmers easier.

We cannot downgrade the role of Lex to that of merely a simple pattern matcher. It finds application in the lexical analysis phase of a compiler too. More details about the construction of a lexical analyser using Lex can be seen in the book 'Compilers: Principles, Techniques and Tools' by Alfred V. Aho, Ravi Sethi and Jeffrey D. Ullman.

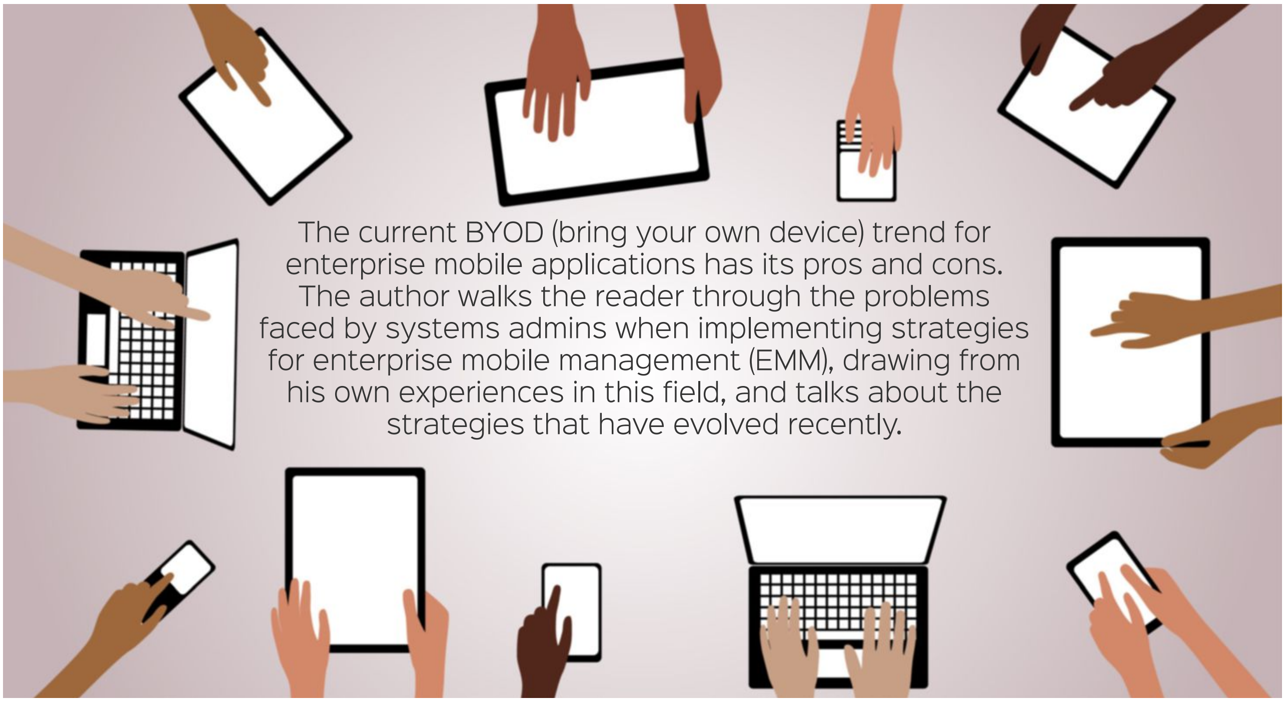
References

[1] *http://dinosaur.compilertools.net/lex/*
[2] *https://en.m.wikipedia.org/wiki/Lex_(software)*

By: Sharon Sunny

The author is an assistant professor at Amaljyothi College of Engineering, Kerala. He can be reached at *ssharon099@gmail.com.*

What Enterprise Mobile Management Strategy Should You Adopt for BYOD?



The current BYOD (bring your own device) trend for enterprise mobile applications has its pros and cons. The author walks the reader through the problems faced by systems admins when implementing strategies for enterprise mobile management (EMM), drawing from his own experiences in this field, and talks about the strategies that have evolved recently.

Developers or product managers of enterprise grade mobile apps often wish their apps could be managed by enterprise IT admins using popular Enterprise Mobile Management (EMM) solutions. Similarly, IT admins are on the lookout for the ideal mobile strategy to manage and secure business apps, with data residing on the user's personal mobile device.

Over the course of my work in this domain, I found solutions to address both these important requirements. It happened while working on an assignment to define an extensible mobile strategy for a popular workforce management (WFM) mobile product. The task was to enable it to be managed by IT admins using popular EMM solutions in the market, whether Air-watch (AW), MobileIron (MI), SOTI or SAP Afaria. The major requirement was to enable the IT admin to dynamically configure WFM to set key properties like the server URL from the EMM admin console.

This article is based on my experience from that exercise. It will provide insights to help you decide the best mobile management strategy for your needs. Additionally, it will help you understand the new, standard EMM integration approach by highlighting the pros and cons of each of the legacy EMM integration approaches normally used.

BYOD: Benefits and trends

The use of mobile devices for work operations has become quite common. In fact, with BYOD (bring your own device), it has become a basic need nowadays.

According to Wikipedia, BYOD refers to the policy of permitting employees to bring personally owned devices (smartphones, tablets, etc) to their workplace, and to use those devices to access privileged company information and applications.

BYOD has resulted in a lot of benefits like:

1. Increased productivity of employees when they work from devices they are familiar with as it helps them complete tasks faster. A survey carried out by Dell and Intel confirms this.
2. Employee satisfaction.
3. Cost reduction—no hardware/COPE (corporate owned personally enabled) device procurement.

Owing to the benefits mentioned, there has been substantial adoption of BYOD in enterprises in the past few years and this is expected to grow even more in the future. According to Gartner, half the employers around the world were operating on the basis of BYOD by the end of 2016, and 90 per cent of organisations will support some aspect of BYOD through 2017.

The need for MDM alternatives for BYOD

Mobile device usage for work has led to IT admins searching for technology solutions to secure these devices and safeguard company/business data.

In the past, mobile device management (MDM) solutions worked well, giving IT admins the management capabilities to manage COPE devices. The usual MDM solution enables IT admins to manage and govern the complete mobile device. MDM allows IT admins to wipe all data, locate devices, apply policies and generally govern COPE devices.

The rise in BYOD opens up the need for alternate mobile management strategies as MDM isn't a fit for BYOD. Let us briefly understand the rationale behind the need for alternatives.

Along with benefits, BYOD comes with challenges. With BYOD, the chances are high that enterprise data on the user's personal device can be compromised. For example, if an employee uses a smartphone to access the data on the company network and then loses that phone, untrusted parties could retrieve any unsecured data from the phone. Such risks call for an appropriate mobile strategy to secure enterprise apps and data.

MDM does not mesh well with BYOD users as they would like to keep their privacy intact while they use their smartphones for work. Users may not like MDM solutions to completely govern their personal devices, which may have their personal files along with business apps and data. Users will be reluctant to use their devices for work if the governing MDM solution is always keeping an eye on their geo location or if there is a chance that their personal files may get accidentally wiped by the IT (MDM) admin.

On the other hand, IT admins may like to at least have basic mobile management capabilities, even for BYOD, so as to effectively do the following:

- *Distribute business applications (in-house or third party) from one place.*
- *Secure data on the move.* Company employees will run business apps on their mobile devices and may fetch business data over public/open Wi-Fi or 2G/3G networks. This confidential data getting transferred over networks while users are on the move is termed as data on the move. It becomes important to take measures to safeguard this data as it can be sniffed and compromised on open networks.
- *Secure data at rest.* Company employees may save business data/files locally on their mobile devices while using business apps. These confidential data files stored locally can be extracted by anyone who gets access to the mobile device. Thus it becomes important to safeguard these data files.
- *Protect data leaks.* There could be a few business mobile apps serving very sensitive data to the end user, like the price quotations of a product. IT admins may want to restrict the screen capture or copy/paste of such sensitive pieces of information from the application.
- *Configure the application.* IT admins may want to set up business apps for their employees by dynamically configuring key parameters like the enterprise backend server URL and port, where the application should connect or fetch data from.

Containerisation and MAM – the MDM alternate strategy best suited for BYOD

The need for the earlier mentioned mobile management capabilities and the partial mismatch between what MDM does and BYOD users want, have resulted in containerisation. This is a new methodology which separates business data from personal data on the user's device. This methodology creates a separate and secure storage space on the device to store business apps and data, away from personal data. This space can be thought of as a separate container/box which keeps business apps and data secure in silos, away from intruders.

Mobile application management (MAM) is also gaining popularity with containerisation. MAM is about managing just the business apps used for business operations instead of managing the entire device. MAM and containerisation go hand in hand, and have become the mobile management strategy for BYOD.

The older EMM integration methodologies for containerisation and MAM

Initially, EMM (enterprise mobility management) vendors devised diverse integration methodologies to achieve containerisation and MAM. These had some benefits as well as quite a few downsides.

Let me highlight a couple of old methodologies along with their pros and cons, which I experienced while doing some WFM MAM work for iOS and Android.

1. ***MAM (EMM) SDK integration methodology:*** In this methodology, the developer needs to integrate the EMM propriety mobile MAM SDK code into the mobile application code. Each EMM propriety MAM SDK library code will be different, and will provide varying mobile management feature sets. There are several EMM solutions in the market and, with this approach, MAM SDK code for each of these EMMs has to be plugged in with the mobile application to support them all. The following are the pros and cons of this approach.

Pros

- Full blown MAM feature support.
- Fine grain management and control.
- Possibility of extended/custom management and security features.

Cons

- Can only support internal mobile apps with MAM SDK. Chances of managing public mobile apps from third party ISVs are less as the app may not have EMM SDK code in it.
- EMM vendor lock-in, if the mobile application is integrated with the MAM SDK code of just one EMM. To support the new EMM, MAM SDK code of the new EMM has to be plugged in within the app code.
- Multiple EMM MAM SDK codes in a single mobile application will increase the following:

a. Code complexity;
b. Application binary—the key consideration is to look at the mobile device storage capacity;
c. Side effects owing to code conflicts, as more or less all MAM SDK codes will be leveraging similar events within the application;
d. Performance degradation of the application.

- Maintenance overheads with constant upgrades for the latest MAM SDK library updates.
- Unavailability of the SDK can become a bottleneck. While working on WFM, initially (Q1, 2015) we integrated an iOS variant with MI's App connect library. For Android, the MI MAM SDK/library wasn't available.

2. ***App wrapping methodology:*** In this methodology, the already compiled and packaged mobile app is wrapped with MAM (EMM) vendor dynamic libraries, and this is called app wrapping. MAM libraries are layered over the already built mobile application binary and then the complete set is recompiled, repackaged and resigned with the EMM app signing certificate to generate a new MAM capable mobile app binary. Post wrapping, standard system calls from the original mobile app are routed through the MAM API library to ensure that the calls are secured and managed. This methodology does not require any development work, that is, no code change is required to hook the MAM SDK. There are several EMM solutions in the market and, with this approach, the mobile application needs to be wrapped with the MAM SDK of all EMMs. The following are the pros and cons of this approach.

Pros

- No development/code change is required.
- Public mobile apps from third party ISVs/developers can be covered as well.

Cons

- Wrapping public apps from third party ISVs/developers or even private apps isn't right and is not recommended. It violates app terms and copyright rules.
- Not a reliable methodology, as it creates a lot of issues and side effects. For WFM, initially (Q1, 2015) we used it for both Android and iOS variants. Post wrapping (with old wrapping engine versions from MI and AW), the app used to get stuck at the landing page with a blank blue screen. Later, after a couple of months with newer wrapping engine versions, the app was able to move ahead from the landing page but used to crash randomly in different modules. On detailed research on Android variants, we found that the MI wrapped library had issues with Implicit Intent handling within the *resolveActivity* method of the PackageManager class from the Android OS.
- It can interfere and obstruct certain functionalities of the app. For WFM Android wrapped with MI, we found that the MI MAM libraries were not allowing the app to fetch GZip data from the server, and there wasn't any way to configure and allow it. This became a big bottleneck and we had to drop this approach eventually.
- Wrapped apps may not support full blown MAM feature sets and will not be able to provide fine grained control like in the SDK approach.
- For WFM Android, post wrapping with MI and AW, we ran into the blocker issue of reaching the 64k method count limit of Dalvik. Android apps run on Dalvik VM (DVM).
- Wrapping with multiple EMM (MAM) libraries created conflicts.
- For WFM, we had to add a different EMM specific code to receive dynamic app configuration from EMM.

Old methodologies could achieve varying levels of containerisation and MAM for small/medium mobile applications but had several downsides as mentioned above. Moreover, the rapidly changing market introduced several methodologies and thus fragmented the market. It created a lot of confusion and chaos amongst IT admins, product owners and app developers to identify the right way to achieve containerisation and MAM.

OS containerisation – standard and recommended EMM integration methods

In the past few years, Apple and Google realised the increasing use of personal mobile devices for work and the need for standardisation. So they took the initiative to bake in containerisation and MAM capabilities right into the mobile OS, that is, iOS and Android (AndroidforWork or AFW). Mobile OS native containerisation can be thought of as a new, standard, universal approach. I will term it as OS containerisation for the rest of this article.

The AppConfig community (a group of EMM providers, ISVs/developers and enterprises) has been formed to standardise and streamline the OS containerisation and integration process. It has come out with an EMM independent methodology to leverage OS containerisation features. As a result, many of the management and security features are automatically taken care of by the OS and will not require any development. For a few features, like app configuration, minimal but standard OS code changes can be made so as to receive dynamic app configuration values from any EMM. With OS containerisation, managed mobile applications can be governed by any app configuration member EMM without any EMM specific code. The following are the pros and cons of this approach.

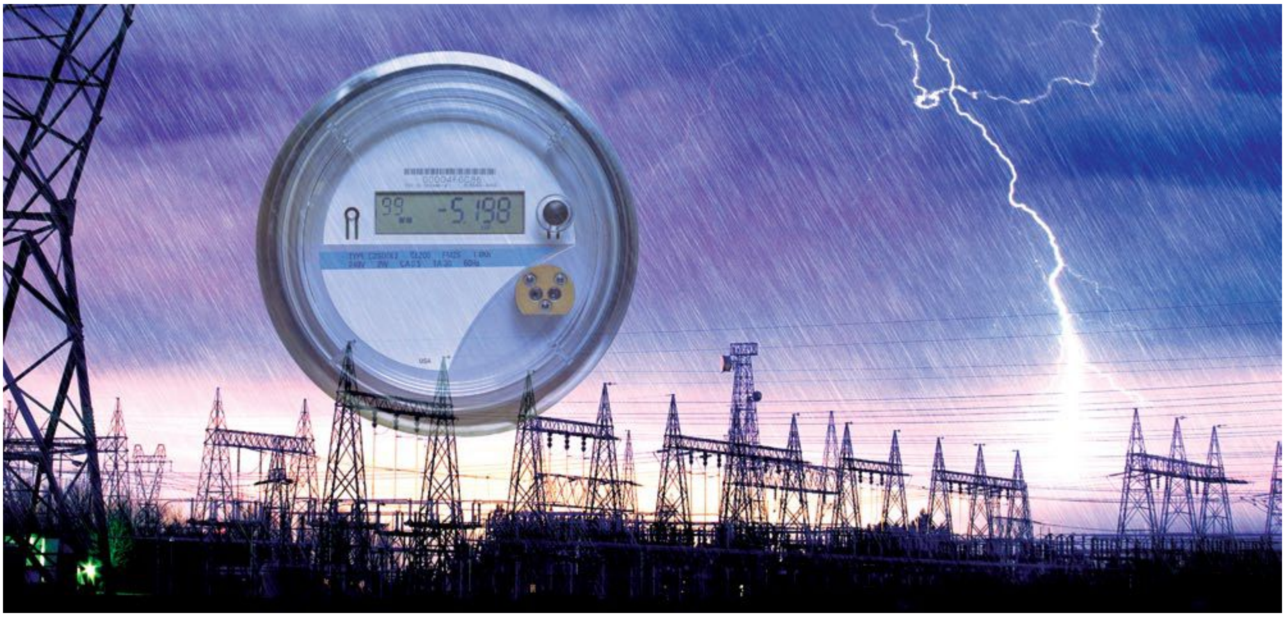
Pros

- Reliable approach, as it is backed by mobile OS vendors (Apple/Google), EMMs, enterprises, ISVs/developers.
- No conflict, nor code redundancy via a single unified, standard mobile OS infrastructure, code for containerisation and MAM.
- No development/code changes are needed.
- Internal (system) apps and public mobile apps from third party ISVs/developers can be covered under this.
- No EMM vendor lock-in, as IT admin can change the existing EMM with another app configuration member EMM, and it will work seamlessly.
- No 64k method count problem on Android.

Continued to page 103...

Gridlabd: Open Source Software for Smart Grid Network Studies

The distribution of electric supply generally faces a few problems. The good news is that these problems can be modelled mathematically and studied without any danger to life and limb. Gridlabd is open source software which enables smart grid studies.



Economising the operation of electricity generators is a problem (called the 'economic dispatch problem') that has been faced since the 18th century. Mathematical models have been developed to study such problems. Another problem is the transmission of power from the generator to the customer, which is called 'optimal power flow'. The third is the 'unit commitment problem', which refers to finding the least-cost ON/OFF state of the available power generation resources to meet the electrical load.

Economising power system operations is still a problem; hence, the term 'smart grid'. A smart-grid is capable of modelling and studying each device connected to the distribution system, which is highly inefficient compared to generation and transmission systems. Therein lies the importance of Gridlabd software. Gridlabd is capable of modelling each and every component in the power system with mathematics running in its core. It is possible to simulate time-series characteristics from micro-seconds to years. Hence, various cases can be studied on a computer before actual system implementation. It helps in improving the efficiency of an existing system and the future expansion of the distribution system. Figure 1 (courtesy Wikipedia) shows a typical power system network.

Gridlabd (GLD)

Gridlabd was developed by the US Department of Energy (DOE) at Pacific Northwest National Laboratory (PNNL) under funding from the Office of Electricity. While writing this article, the source code for Gridlabd was not available in Ubuntu (am presuming you're a Debian user). So the available Red Hat package manager (rpm) has been converted to Debian (Deb). Gridlabd is the first simulation platform in which modern energy systems are inbuilt. Time series simulation and load modelling become easier with Gridlabd from sub-station to customers (residential load). Gridlabd is an open source tool freely available to anyone. It encourages collaboration with industry and academia. The BSD-style licence allows us to add or extract our own modules without compromising the internal intellectual property. The Web portal *http://www.gridlabd.org/gldportal* has a GUI for quick real-time hands-on operation of Gridlabd.

Installation and how it works

Since Gridlabd is not readily available for installation in Ubuntu, we need to convert the available rpm package to Deb. The steps to be followed are given here.

Open a terminal and download the package

```
hithu@linux:$ wget https://sourceforge.net/projects/
gridlab-d/files/gridlab-d/Last%20stable%20release/
gridlabd-3.2.0-1.x86_64.rpm.
```

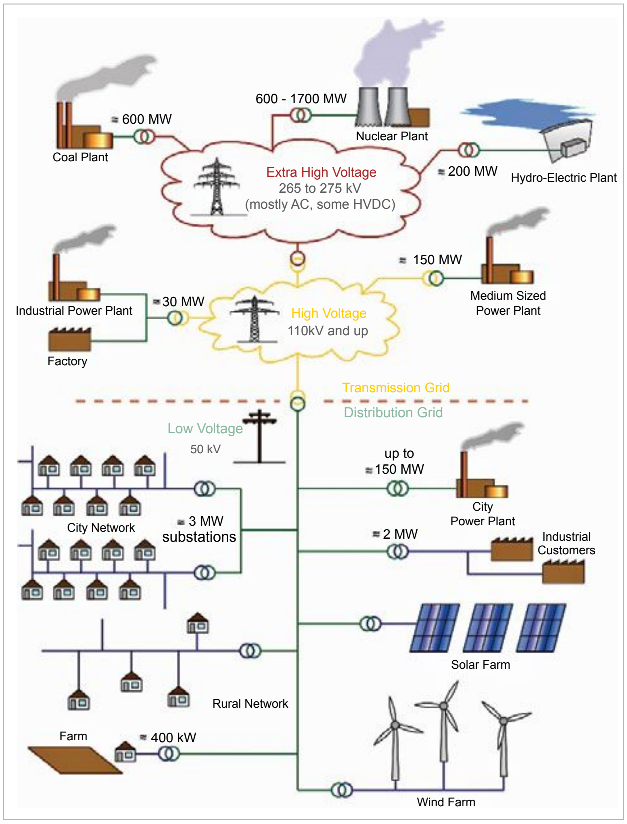


Figure 1: Typical power system network

After downloading, you can check the downloaded file in your home directory. To convert it to a Deb file, you need to install the alien package, as follows:

```
hithu@linux:$ sudo apt-get install alien
```

Convert the rpm to Deb by using the command given below:

```
hithu@linux:$ alien gridlabd-3.2.0-1.x8664.rpm
```

Exit the terminal. Right-click and open the newly generated Debian file using the Ubuntu software center package manager to complete the installation.

Check the installation by using the following command:

```
hithu@linux:$ gridlabd --version
GridLAB-D 3.2.0-5368 (Jojoba) 64-bit LINUX RELEASE
```

To try a simulation of residential temperature variations for a year, save the code given below as *residential.glm* using a text editor (gedit):

```
clock {
  starttime '2017-02-01 00:00:00 UTC';
  stoptime '2018-02-01 00:00:00 UTC';
}
module residential;
module tape;
object house {
  object recorder {
    property air_temperature;
    file temperature.csv;
  };
}
```

In terminal, run the file:

```
asgridlabd residential.glm
```

To see output:

```
more temperature.csv
2017-02-01 00:00:00 UTC,+72.4767
2017-02-01 01:00:00 UTC,+73.5292
2017-02-01 02:00:00 UTC,+74.1789
2017-02-01 03:00:00 UTC,+74.7188
2017-02-01 04:00:00 UTC,+75.1863
2017-02-01 05:00:00 UTC,+75.6680
2017-02-01 05:24:12 UTC,+76.0000
2017-02-01 05:29:10 UTC,+73.9968
2017-02-01 06:00:00 UTC,+75.9057
2017-02-01 06:02:18 UTC,+76.0006
2017-02-01 06:07:40 UTC,+73.9994
2017-02-01 06:25:12 UTC,+76.0006
2017-02-01 06:30:33 UTC,+73.9961
2017-02-01 06:48:21 UTC,+76.0001...
```

Hence, you have the result of temperature variations in a typical house with hourly intervals, for a year. It is possible to change the interval, add the location of meteorological data, measure various parameters, add more implicit devices, and much more.

There are many modules available within this software:

```
gridlabd --modhelp residential
```

The module given above the list of the various classes and objects possible in a residential simulation. For example, class 'freezer' has…

```
class freezer {
     parent residential_enduse;
     class residential_enduse {
       loadshape shape;
       enduse load; // the enduse load description
       complex energy[kVAh];
// the total energy consumed since the last meter reading
       complex power[kVA];
// the total power consumption of the load
```

```
       complex peak_demand[kVA];
// the peak power consumption since the last meter reading
       double heatgain[Btu/h];
// the heat transferred from the enduse to the parent
       double cumulative_heatgain[Btu];
// the cumulative heatgain from the enduse to the parent
double heatgain_fraction[pu];
// the fraction of the heat that goes to the parent
double current_fraction[pu];
// the fraction of total power that is constant current
double impedance_fraction[pu];
// the fraction of total power that is constant impedance
double power_fraction[pu];
// the fraction of the total power that is constant power
double power_factor; // the power factor of the load
complex constant_power[kVA];
// the constant power portion of the total load
complex constant_current[kVA];
// the constant current portion of the total load
complex constant_admittance[kVA];
// the constant admittance portion of the total load
double voltage_factor[pu];
// the voltage change factor
double breaker_amps[A]; // the rated breaker amperage
set {IS220=1} configuration; // the load configuration options
       enumeration {OFF=4294967295, NORMAL=0, ON=1} override;
       enumeration {ON=1, OFF=0, UNKNOWN=4294967295} power_
state;

     }
```

```
       double size[cf];
       double rated_capacity[Btu/h];
       double temperature[degF];
       double setpoint[degF];
       double deadband[degF];
       timestamp next_time;
       double output;
       double event_temp;
       double UA[Btu/degF*h];
       enumeration {ON=1, OFF=0} state;
}
```

Further detailed documentation on power flow analysis is given on the website *http://gridlab-d.sourceforge.net/wiki/index.php/Powerflow*. More technical help on the software can be accessed at *https://sourceforge.net/p/gridlab-d*.

Accurate modelling of distribution networks and connected devices is possible with this open source software. The only drawback is that the original software doesn't have a GUI. It can be used as a benchmark for comparison between various systems. Results obtained can be plotted and interpreted using other open source software like Scilab, LibreOffice Calc, etc. Kindly refer to my article in the February 2017 issue of *OSFY*. As of now, Gridlabd has the flexibility for smart-grid simulation.

By: Hithu Anand

The author is an EEE research scholar, affiliated to Anna University, Chennai. You can contact him at *hithuanand@gmail.com*.

Continued from page 100...

Cons

- May take some more time to mature, as OS containerisation is gradually evolving and a set of new features is getting introduced with every new OS version.
- AFW is complex to set up as it involves tie-ins with several Google consoles and EMM in use to set up the entire system. Examples of consoles are Google Admin Console, Google Play for Work, Work Profile on Android device, etc.
- AFW can be costly as it is powered by Google, which will levy charges on a per-user, per-month basis. These Google charges will be additional costs if your organisation is already paying and using some EMM tool.
- For AFW, GCDS or Google Cloud Directory (earlier known as GADS) may require to be set up to sync your organisation's Active Directory user/group with it. This may not be acceptable as per the policies of many enterprises.

OS containerisation empowers IT admins with the following key mobile management and security capabilities for BYOD:

1. Securing data on the move via app tunnel/per app VPN.
2. Securing data at rest via complete encryption.
3. Mobile app level DLP (data leak protection) by disabling screen capture, disabling copy/paste, selective app wipe, and pin protection for business app access.
4. Single sign-on.

I strongly recommend the OS containerisation integration methodology as it is provisioned by mobile OS vendors Apple and Google and has become the standard, thanks to the participation of popular EMM vendors, ISVs, enterprises, etc. It enables coverage of a wide set of mobile applications in a standardised manner without getting locked with any single EMM vendor solution, and that too in a standardised manner.

By: Gaurav Vohra

The author is a technical/enterprise mobility architect at Impetus Technologies and has more than 12 years of experience in the IT industry. He has been predominantly handling software products and services in the enterprise mobility space. He can be contacted at *gaurav.vohra@impetus.co.in*.

TIPS & TRICKS

An alternative for the *dd* command

We are comfortable using the *dd* command in our day to day administration of computers and servers. An alternative to it is the *dcfldd* command, which also shows a progress report of the process. *dcfldd* is an enhanced version of *dd*, developed by the US Department of Defence Computer Forensics Lab.

To use *dcfldd*, we first need to install it, for which you can use the following command:

```
$sudo apt-get install dcfldd

$dcfldd if=<source> of=<destination>
```

For example:

```
$dcfldd if=/dev/zero of=/dev/null
```

The output is:

```
425216 blocks (13288Mb) written.
```

—***Rupin Puthukudi, rupinmp@gmail.com***

A beginner's guide to Curl

The *Curl* command is used to transfer data from or to a server. It is used with the following protocols: HTTP, HTTPS, FTP, SFTP, TFTP, DICT, TELNET, LDAP, IMAP, SCP, SMTP and SMTPS.

Curl can be used inside the scripts. When you download using the *Curl* command, you can easily pause or resume the download. There are 120 command line options with the *Curl* command. It supports cookies, forms and SSL, multiple uploads with a single command and IPv6.

For example:

```
$curl http://www.my.org
```

The above command shows the contents of this website in your terminal. If you want to save the output of the Curl command, you can use the redirect symbol > or the *-o* option.

For example:

```
$ curl http://www.centos.org > file.txt
```

…or:

```
$ curl http://www.centos.org  -o file.txt
```

Now let's look at how to download multiple files using the *Curl* command:

```
$curl -O http://ftp.open.com/pub/axt.html -O  http://ftp.gnu.
com/pub/bxy.html -O   http://ftp.lkj.com/pub/c.html
```

An important fact to note here is that Curl will use the same TCP connection to download multiple files. This is done to improve the download speed. When we use *Curl -O* (uppercase O), it will save the contents in the file with the original file name in the local machine. If we use *-o* (lowercase o), then we need to give the file name in the command.

Curl is automatically redirected to a new URL if we use the *-L* option.

For example:

```
$curl  -L google.com
```

After running the above command, you can cancel the downloading process using *Ctrl+c*, and then if we want to resume downloading from the last point, use the *-C* option.

```
$curl -C - -O http://ftp.local.com/pub/test.gz

** Resuming transfer from byte position 28609

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

To limit the data rate in the *Curl* command, use the *--limit-rate* option, as follows:

```
$ curl --limit-rate 1000B -O http://ftp.local.com/pub/test.gz
```

To download a file if it is modified before/after a given time, use the *-z* option, as follows:

```
$ curl -z 28-Nov-11 http://www.my.com/aa.html
```

To download *yy.html* from the server that's been modified after November 28, use the following command:

```
$ curl -z -28-Nov-11 http://www.my.com/bb.html
```

Sometimes a website may ask for your user name and password when you wish to view the contents, at which point, pass the user name and password in the command as shown below:

```
$curl -u username:password URL
```

This can be very useful when downloading files from password protected FTP sites.

To ignore the SSL certificate, use the *-k* option with the following command:

```
$curl -k https://www.abc.com
```

There are many other uses for Curl. You can check out the man pages for more details.

—Ajay Trivedi, ajay.trivedi@minjar.com

Analysing your Wi-Fi signal strength on Ubuntu

The Wi-Fi signal strength can vary from room to room, so it's good to know which is the best spot in the house or office to work from. Here is a simple script to find out the signal strength in Ubuntu.

Create a text file called *signalstr.sh* and enter the following script:

```
#!/bin/bash
echo "Please enter the wireless interface you wish to use"
read wifiinterface
echo "Please enter the routers ip address(or any system you
wish to ping)"
read pingaddress

count=0
echo "Interface: $wifiinterface    Ping Address: $pingaddress"
>> ~/wifianalyse.log
echo "     Time     Count   Output  Ping"
echo "     Time     Count   Output  Ping" >> ~/wifianalyse.log
while [ 1 -eq 1 ]
do
      sleep 1
      count=$((count + 1))
      countf=$(printf "%05s" $count)
wifioutput=$(iwconfig $wifiinterface | grep -i --color quality)
      pingoutput=$(ping -c 1 $pingaddress | grep -E -o -m 1
'time.{0,10}' & )
      curtime=$(date +%d/%m/%y\ %H:%M:%S)
      echo [$curtime] "| " "$countf""  |  " $wifioutput " |   "
$pingoutput
      echo [$curtime] "| " "$countf""  |  " $wifioutput " |   "
$pingoutput >> ~/wifianalyse.log
done
```

Save the text file and then run the script using the following command:

```
#sh signalstr.sh
```

After running this script, you're instantly asked to enter the wireless interface name you wish to test, along with the IP address of your router to ping. This information is crucial to the running of the script, so make sure you enter the correct information. As soon as you enter the two details, it will start showing the signal level for you.

—Harish Tiwari, harishtiwary46@gmail.com

Disabling a guest account

Guest accounts on a computer can be useful, but some people (myself included) see them as a waste of space. To disable the guest account, run the command *sudo*.

Open the terminal and install as follows:

```
$sudo apt install gksu
```

Now run the following command:

```
$gksudo gedit /etc/lightdm/lightdm.conf.d/50-no-guest.conf
```

Add the following command and exit:

```
[SeatDefaults]
allow-guest=false
```

This is a simple configuration file modification, which the system reads during each boot up.

—Harish Tiwari, harishtiwary46@gmail.com

END

Share Your Linux Recipes!

The joy of using Linux is in finding ways to get around problems—take them head on, defeat them! We invite you to share your tips and tricks with us for publication in *OSFY* so that they can reach a wider audience. Your tips could be related to administration, programming, troubleshooting or general tweaking. Submit them at *www.opensourceforu.com*. The sender of each published tip will get a T-shirt.

DVD OF THE MONTH

Secure your network with this toolkit

In case this DVD does not work properly, write to us at support@efy.in for a free replacement.
antergos Live
for everyone 17.2
A modern, elegant and powerful operating system based on Arch Linux. It is fully configured with sane defaults that you can use right away
Recommended System Requirements: P4, 1GB RAM, DVD-ROM Drive
March 2017
DVD
Note: Any objectionable material, if found on the disc, is unintended, and should be attributed to the complex nature of Internet data.
Network
Security Toolkit (NST24)
A Fedora based live Linux that provides easy access to best-of-breed open source network security applications
CD Team e-mail: cdteam@efy.in

Network Security Toolkit (NST 24)

This is a Fedora based live Linux operating system that provides easy access to best-of-breed open source network security applications. The main intention of developing this toolkit was to provide the security professional and network administrator with a comprehensive set of such tools. You can find the live, bootable ISO image of NST 24 in the *other_isos* folder on the root of the DVD.

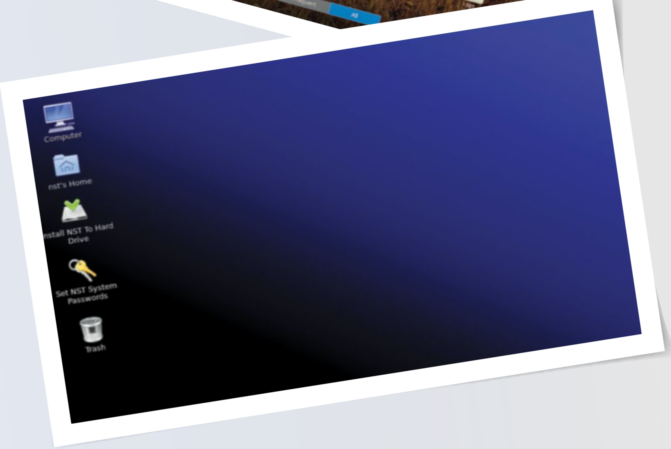
Antergos 17.2

This is a modern, elegant and powerful operating system based on Arch Linux. It provides a fully configured OS with sane defaults that you can use right away. Users need not be Linux experts or developers in order to use Antergos. It's a Linux distro for everyone. The bundled DVD can boot in live mode and run Antergos on any 64-bit supported computer.

What is a live DVD?

A live CD/DVD or live disk contains a bootable operating system, the core program of any computer, which is designed to run all your programs and manage all your hardware and software.

Live CDs/DVDs have the ability to run a complete, modern OS on a computer even without secondary storage, such as a hard disk drive. The CD/DVD directly runs the OS and other applications from the DVD drive itself. Thus, a live disk allows you to try the OS before you install it, without erasing or installing anything on your current system. Such disks are used to demonstrate features or try out a release. They are also used for testing hardware functionality, before actual installation. To run a live DVD, you need to boot your computer using the disk in the ROM drive. To know how to set a boot device in BIOS, please refer to the hardware documentation for your computer/laptop.



R N I No. DELENG/2012/49440, Mailed on 27/28th of Advance month
Published on 27th of Advance month
Delhi Postal Regd. No. DL(S)-01/3443/2016-18
Licenced to Post without Pre-Payment Licence No U(S)-56/2016-17

Loonycorn

Our Content:

- The Complete Machine Learning Bundle
  10 courses | 63 hours | $39
- The Complete Computer Science Bundle
  8 courses | 78 hours | $39
- The Big Data Bundle
  9 courses | 64 hours | $45
- The Complete Web Programming Bundle
  8 courses | 61 hours | $41
- The Complete Finance & Economics Bundle
  9 courses | 56 hours | $49
- The Scientific Essentials Bundle
  7 courses | 41 hours | $35
- ~15 courses on Pluralsight
  ~70 on StackSocial
  ~60 on Udemy

About Us:

- 50,000+ students - most of them happy:-)
- <10 team - all of us happy:-)
  ex-Google | Stanford | IIM-Ahmedabad | INSEAD