

DevOps Intelligent Networking with Automated Deployment in Linux

Pushendra M. Chavan, Sandeep P. Abhang, U. B. Shinde



Abstract: DevOps shortly refers to the common intelligent practice of combining the Software Development and Operations to carry out certain set of tasks for automation of jobs. Methodologies in DevOps have become first choice for the system administrators to carry out operations [3]. Deployment of the Operating System, since a big task, can be carried out by many ways, like manual or an automated method [1]. Manual method is a time consuming and prone to errors whereas the automated method is much better in terms of management and repetition. This research paper covers the need of such Automated Deployment method including much needed tasks such as server and client side configurations, centralized monitoring, logging and software patch management. As an addition, the proposed network in itself is self healing at the Operating System level, without compromising security and removing corruption at libraries and binaries level. Use of this optimized method helps reducing administration effort and is prone to no errors. A Self healing service makes sure of no corruption for long term and centralized logging helps monitoring the network as well [5]. The Intelligent Networking for Automated Deployment is a DevOps reference project which can be useful for automation in the Organizations for heavy and uniform deployments along with additional monitoring configurations.

Index Terms: network automation, intelligent networks in Linux, centralized logging, kickstart deployment, computer security, computer network management.

I. INTRODUCTION

DevOps is a common practice of making fusion of Software Development along with the practice of Computer Operations. Generally, this isn't associated with any specific technology but brings many technologies together, and is completely platform independent. In Linux Administration, DevOps refers to the use of Open Source Development Practice [2] along with day to day operations to carry out complex tasks, including deployment, monitoring, patch management and security measurement. It also includes configuring services like web services, networking, load balancing, and clustering & storage management.

These tasks are very complex in nature, so always need an expert running them, which involves cost, so automation in all these services and client-server management makes it easy to handle and inject updates in the process throughout its lifecycle. As per the Industry Standard, a Client-Server or a Domain Network model comprises of many computer systems connected to each other using Layer 1/2/3 devices. These systems interact with each other through many services; one or many of these systems can act as servers providing many services like Domain Naming Service, Dynamic Host Configuration Protocol Services, Remote Logging Services, Web Services and many more. It is necessary to have a distributed load of services on many dedicated servers and also there is a need of a backup in case one fails to answer to a service request. This setup can be called as a Load Balancer, High Availability[2] or even High Performance Computing. Though most tools used in Linux are free and Open Source, one needs to have an expertise to handle those effectively. The expert logic can build automation software by making all these services available, deployable and self healing, making it really an Intelligent Network.

II. RELATED WORK

Automation is the need of today's task force which helps perform many activities. There are many preexisting software which perform automated tasks using open source programming languages using perl/python through ansible automation or puppet whereas automatic deployment is out of scope from these automation software, so software like Red Hat Satellite or HP provided HPSA, or Advanced Intelligent Deployment by third party vendor can be used along with cobbler, katello or anaconda. Software like these are very much costly in nature and require specialized talent/certified engineer to handle, whereas this paper explains the custom method of automating such builds. There are only little vendor software available online as follows.

A. Red Hat Satellite

Red Hat provides a product named Red Hat Satellite[1] which consists of a function named katello which can be used for Provisioning but Red Hat officially supports build of Red Hat Enterprise Linux, not even another Operating system like CentOS or Fedora or Scientific Linux, This product is again a very costly product only to be used in larger organizations.

B. Cobbler

Cobbler is an Open Source tool which is used to build ISOs using kickstarts [3] and custom deployments.

Revised Manuscript Received on 30 July 2019.

* Correspondence Author

Mr. Pushendra M. Chavan*, MTech Student, Dept of CS & E, Chh. Shahu College of Engineering, Aurangabad, India.

Prof. Sandeep P. Abhang, Professor & HoD, Dept of CS & E, Chh. Shahu College of Engineering, Aurangabad, India.

Prof. U. B. Shinde, Principal, Chh. Shahu College of Engineering, Aurangabad, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

This product is free of cost but has few limitations and needs to be first deployed on Linux/Windows machine in order to function. It can build the systems with user defined kickstart files but can't render its own kickstart files. The limitation of cost and custom kickstarts can be taken care in this methodology and administrators can build machines as per their requirements.

III. IMPLEMENTATION

The proposed system will be in the form of an rpm package which can be deployed on any Red Hat Enterprise Linux, CentOS, Fedora, Oracle Linux or Scientific Linux machine. The said package will deploy few functionalities like

- Scanning the said path looking for the master DVD media for installation.
- Will ask administrator if THE BUILD IS FOR MASTER SERVER or client build.
- Requests to provide additional details like services and there configurable parameters.
- Will build a resultant ISO media which will be self sustaining for deployment and configuration.
- The software management system for updates.
- Intelligent self heal for any corruption.

So the provided server will be self deployed and will provide services like dynamic host configuration protocol network configurations, IP/Host MAC ID binding, pre-execution deployment services, web services and most importantly, the remote logging to the server which will be used for monitoring purposes. The subsystem also contains a functionality named self heal which will scan the systems whether server or client on a timely basis.

The proposed system follows a three step process.

Deployment of Master System: This system will be either a personal laptop or an industry standard box running Red Hat Enterprise Linux [1], CentOS or Oracle Linux. The master software will be deployed on this system which is provided in the rpm format. Deployment will BE PERFORMED using yum command along with ITS dependencies.

Spinning the Server Build: Post deployment of the master software, the second step will be running/executing the master software to build a custom deployment media for the actual server in the network along with all customizations. This built Software ISO will be shipped to the requested team for further deployment.

Deploy the actual Server: This will be the last process, where the actual server will be deployed using the Master Software Product ISO and will setup the complete network by itself. The client deployments will take place using pre execution environment. A complete subsystem with pre packaged services will be configured here and will be ready to be in production.

IV. BASIC ARCHITECTURE & FLOW DIAGRAM

There are three stages of the main process, which includes the base system process, involving the host Operating system where the master software gets deployed. The host operating system needs to have a basic Operating system over the top of the basic hardware, kernel, shell and shared libraries. The second phase is to actually spin the media whereas the third phase is to deploy the multifunctional

server which performs monitoring and self healing operations.

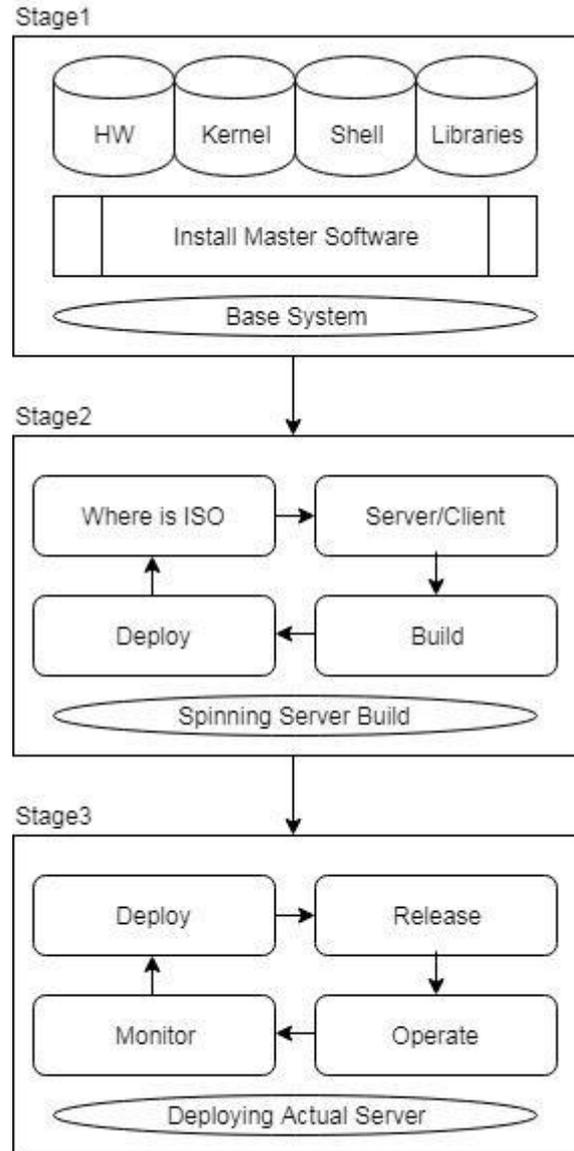


Fig. DevOps Intelligent Networking with Automated Deployment in Linux

V. PERFORMANCE MEASUREMENT

Use of the DevOps practice in Networking and Automated Deployment in Linux makes it faster in execution, easier in handling, reliable in accountability, smooth in performance and accurate in self healing. The comparison of 1, 2, 5 and 10 system's performance measurement table is as follows.

Feature	No of systems	01	02	05	10
Self Healing on occurrence		311	311	311	311
Automated Deployment	PXE	265	412	556	600
Remote Logging & Monitor		Live	Live	Live	Live

*all figures in seconds

Another performance comparison with manual method and automatic method is as follows.

Feature	Manual	DevOps	Computing. Date of Xplore.
Self Healing on occurrence	No Defaults	311 (crontab)	06 January 2003
Automated PXE Deployment	9000 (approx)	265 (no user input)	3. M Rajkumar, Anil Kumar Pole, Vittalraya Shenoy Adige & Prabal Mahanta's "DevOps culture and its impact on cloud delivery and software development", published in IEEE Xplore. Date of Conference: 8-9 April 2016 Date Added to IEEE Xplore: 29 September 2016 Electronic ISBN: 978-1-5090-0673-1 Print on Demand (PoD) ISBN: 978-1-5090-0674-8 INSPEC Accession Number: 16342034 DOI: 10.1109/ICACCA.2016.7578902
Remote Logging & Monitor	No Defaults	Live	4. Matt Callanan & Alexandra Spillane's "DevOps: Making It Easy to Do the Right Thing", published in IEEE Software (Volume: 33 , Issue: 3 , May-June 2016)
Server Config	No Defaults	Part of Deployment	5. Johannes Wettinger, Vasilios Andrikopoulos, Frank Leymann & Steve Strauch's "Middleware-Oriented Deployment Automation for Cloud Applications", published in: IEEE Transactions on Cloud Computing (Volume: 6 , Issue: 4 , Oct.-Dec. 1 2018)

*all figures in seconds

VI. CONCLUSION

Use of bash programming is used to build up the software from scratch, shipped via MIT General Public License, completely Open Source, free of cost. All industry standards and requirements are fulfilled. Light weight, ready within few minutes to generate number of ISOs and free to modify. Use of traditional protocols like hypertext transfer protocol, file transfer protocol, trivial file transfer protocol, dynamic host configuration protocol and smartest open source tools like genisoofs [1] and bash makes life easy for the administrators here.

System like this has its own advantages/pros and disadvantages/cons.

Pros:

1. Easy to setup and light weight.
2. Open Source, Free, built from scratch, free to modify and use.
3. Zero human interaction required for server deployment/configuration and maintenance.
4. Self Heal makes it much better in order to avoid file level corruption.
5. Uniform deployment all across the network.
6. Zero chances of human error as no human interaction needed for Server Deployment.

Cons:

1. Can't figure out Filesystem level Corruption.
2. Can't stop intrusion on the server.
3. Won't take user data backup automatically.
4. Additional Configurations are required to achieve this functionality.
5. Available only for Linux as of now. Not for non-Linux platforms.

VII. ACKNOWLEDGEMENT

Our Thanks to the Open Source Community for GNU, Anaconda, Fedora, Cobbler and genisoimage for providing these utilities under GPL/MIT license and putting all efforts building these utilities. Without these utilities, the resultant system for DevOps Automation wouldn't have been possible. A Special Thanks to Red Hat and Red Hat Technologies for providing all possible sources for making this platform ready for deployment.

REFERENCES

1. https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux [online] Red Hat's online documentation guide to customize installation media, and downloads for Proof of Concept Deployment.
2. M.J. Katz, P.M. Papadopoulos & G. Bruno's "Leveraging standard core technologies to programmatically build Linux cluster appliances", published in: Proceedings. IEEE International Conference on Cluster

AUTHORS PROFILE



Mr. Pushpendra Chavan is pursuing Master of Technology from Chh. Shahu College of Engineering, Aurangabad. He has published around 5 international journal papers on diskless client and has filed a patent on the same which is under consideration by Patent Authority, Government of India. Along with studies, Pushpendra is working as a Principal Technical Support Engineer at Red Hat India Private Limited and is a Red Hat Certified Architect.



Prof. Sandeep Abhang is working as a Head of the Department for Computer Science and Engineering in Chh. Shahu College of Engineering, Aurangabad. He is an active contributor and a promoter of Open Source Software Community. Professor Sandeep Abhang has expertise in Storage, Networking and Linux. He has organized many Open Source Events in Chh. Shahu College of Engineering, Aurangabad to promote free, alternative & better choices of Open Source software as compared to licensed products.



Prof. U. B. Shinde pursued bachelor, Master and Ph.D. degree in Electronics and Telecommunication, from Dr. B. A. Marathwada University, Aurangabad. He is working as a Principal at Chh. Shahu College of Engineering, Aurangabad. Professor U. B. Shinde is the mind behind Open Source Labs in Chh. Shahu College of Engineering where students from all streams can go and make use of the Software library.