2020 IEEE International Conference on Convergence to Digital World – Quo Vadis (ICCDW 2020)

Autonetics and Administration for IT Laboratories

V. Karthikeyan
Department of Information Technology
A.P. Shah Institute of Technology
Thane, India
eyankarthik23@gmail.com

Uddhabendra Maity
Department of Information Technology
A.P. Shah Institute of Technology
Thane, India
u.b.maity@gmail.com

Atharv Shetty
Department of Information Technology
A.P. Shah Institute of Technology
Thane, India
atharv32@gmail.com

Sameer Nanivadekar
Department of Information Technology
A.P. Shah Institute of Technology
Thane, India
ssnanivadekar@apsit.edu.in

Vishal Badgujar
Department of Information Technology
A.P. Shah Institute of Technology
Thane, India
vsbadgujar@apsit.edu.in

***Abstract*—Majority of IT labs in today's academic institutions face operational issues in the management of multiple systems simultaneously. The best example would be when a particular software needs to be installed in the labs, it becomes a tedious and time consuming process for the lab assistant to manually install the software in each and every system in the lab. Also in some cases where the students forget to shutdown their respective computers, it becomes the responsibility of the lab assistant to shutdown the PCs manually. These challenges cause lack of access control and inadequate security. Moreover, there is lot of work pressure which leads to sub-optimal work schedules. To keep track of access records of the systems, we would also be designing a web-based GUI which records and displays the access information of PCs too. Thus a well thought out design for automating these various operational tasks inside the IT lab environments has been implemented and worked upon. In this paper we've discussed the roles and benefits of automating the IT labs for the students as well as lab instructors.**

Keywords—ansible, deployment, automation, administration, installation

I. Introduction

As we are using a free and open-source platform for our purpose, many labs can be automated using the same architecture at a very feasible price. Ansible provides the automation of IT infrastructure which includes – creation of virtual machine, installation of new softwares, Docker or Kubernetes Deployment and Configuration in VM with the help of various configuration files. Ansible also provides different types of modules such as custom modules and core modules. There is no need of any knowledge about programming languages to work with ansible. Ansible does its work like a professional if customized with proper facts and experience. The goal of Ansible is to implement automated application deployment, automated configuration management, automated continuous delivery [7]. Ansible works over SSH, ensure that the target Machine or Server is accessible over SSH.
It supports all type of SSH authentication.
1.Username/Password Based SSH Auth
2. Public/Private Key Based SSH Auth
3. Tunnel and Jump box based SSH Auth

II. Literature Survey

Xavier Decoster [1] proposed an architecture stating the usefulness of NuGet which is an open source visual studio extension. He named the architecture as ProNuget. ProNuGet gives a solid, practical knowledge of both how to maintain the software dependencies under control and a long term reliability. Doesn't matter if you're coding entirely with .NET framework assemblies or also using HTML,CSS and JavaScript files in your various applications, the featured system shows you how to manage their requirements reliable and smoothly.

D. Palma and T.Spatzier [2] presented a meta model for stating various IT services. This meta model states both the overview of the structure as well as the managing sector. A Topology Template states the overview of the service. Plans states the various processing models that are used to build as well as kill a service and also helping in managing a particular service. Node Types are introduced differently for re usability purposes. Service Templates can be based on and built on-top of other Service Templates based on the concept of Requirements and Capabilities introduced in the previous section. This paper helped us in analyzing the different network topology which we can consider for our

978-1-7281-4635-5/20/$31.00 ©2020 IEEE

purpose. Also it helped us in gaining an overview of integrating cloud technologies in the system.

Pavel Mašek Martin and ŠtůsekJan Krejčí [3] proposed their design using the ansible framework where they used ansible as a core part of the design to automate the IT operations to a certain extent. However the utilization and application of this design in real life scenarios was not depicted clearly. This design supported many operating systems at the same time. Thus creating an OS friendly environment. Types of operating systems included : (i) Windows 7; (ii) Windows 10; (iii) Ubuntu 16.04; (iv) Debian 8.It also included the ability to install and uninstall various applications such as Google Chrome, Mozilla Firefox and Microsoft Visual Studio. In addition, several system operations must be enabled: (i) Displaying information and status of stations; (ii) restarting and shutting down stations; (iii) creating users; (iv) creating and deleting a file; (v) installing system updates. As this was the closest and the most relevant research related to our paper, we formed this as our base paper and carried all the further research and development over this paper.

Nishant Kumar Singh [4] from Amity University proposed a very clear and precise description of automated-provisioning also called as self-service provisioning. In this paper he explained the ways of deploying IT services by using predefined procedure that are carried out electronically without requiring human intervention. In a traditional setting, provisioning is a manual process that requires the assistance of several people in several roles and involves multiple steps. It could take days or even weeks to move a request from the submission phase through the actual activation of service. Automating provisioning allows customers to set up and make changes to services themselves by using a Web browser or other client interface. It can provide a more efficient and rapid response to business requests and cut service activation or service change time down to hours or even minutes. This paper helped us in understanding the several ways of deploying the IT services and carry out the operational processes.

J.O.Benson and J. J. Prevost, P. Rad [5] presented an efficient way of taking advantage of scripting techniques. It allows the flexibility of installing all of the necessary and desired software, including updates. Likewise, scripting can be dynamic enough to distribute custom files to each of the N-nodes to ensure that necessary files work appropriately, something that is impossible when using imaging for server set up. An example of this might be distributing lists of IP addresses, custom host-names, or specific license files for software, or the system, to operate correctly. Another advantage of scripting is that it requires very little space. As we are currently designing the whole system in linux environment, we extracted a lot of knowledge about shell scripting from this paper and thus trying to utilize this technique in further operations.

Jay Shah and Dushyant Dubaria [6] in their paper "A Survey of DevOps tools for Networking" researched and stated various facts about using the Redhat's open-source technology “Ansible” . For ansible to work, a proper working version of python is required as it requires the various libraries from python. Ansible works in two modes: ad-hoc modes and playbook files. The ad-hoc mode involves passing the instructions to the remote servers via ansible commands. Using the other mode i.e. with the help of playbooks, all the tasks needed to be mentioned in the playbook yml file. Once the playbook is executed all the tasks are performed accordingly.

III. EXISTING SYSTEM ARCHITECTURE

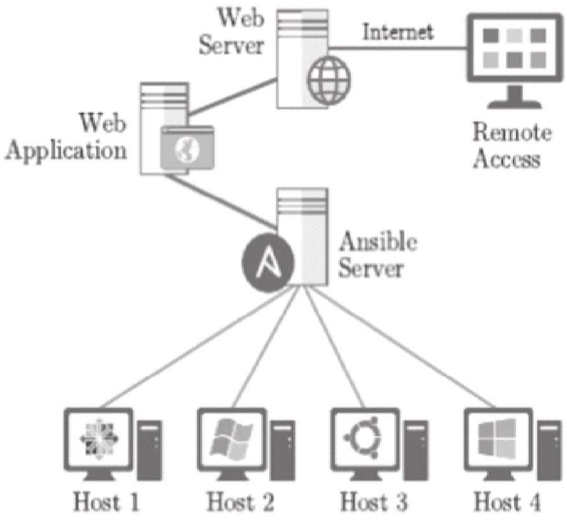

Fig. 1. Ansible architecture where several hosts are connected with the ansible server.

In the above figure, the administrator is provided with remote access over the internet and is connected to the web server which is responsible for hosting the web application. In the back end of that web application ansible server is integrated, and this web application will connect to all the host PCs within the network and also support multiple operating systems. Ansible is easy to implement as it doesn’t require installing any kind of installations on the client machines, because it uses SSH (Secure Shell) to push the configurations[10]. Ansible automation Jobs are added to CI server to initiate deployment to staging and production environments [9].

Remote Access: The Administrator will be provided with the PC, where he will have access to all the PCs which are connected in network. These systems should be connected to internet via the LAN.
Web Server: Web server is responsible for hosting the web application on the respective systems.

IV. PROBLEM STATEMENT

In current university labs most of the administrative tasks are done manually which consumes lot of time and efforts. With the help of ansible framework and a proper supporting GUI, we can unleash and maximize the full potential of the servers and many of the current lab administrative problems can be resolved easily.
The core functionality of this architecture is automated installation and management of software packages. Ansible is originally only a command-line interface tool and thus it

lacks an elegant user interface. Only a well versed user will be able to operate on a command line tool, which means a layman user will find it difficult to operate and run the ansible commands effectively. This is done by means of so-called playbooks defined in YAML[8], which describes host configurations

V. Proposed System Architecture

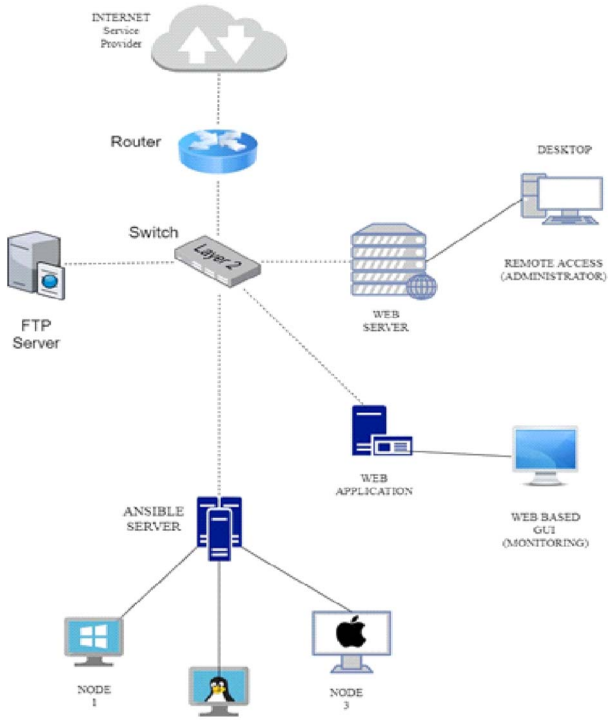

Fig. 2. Proposed System Architecture

These are the primary objectives of our proposed system architecture:

- To Automate the Software Package installation and upgrading process.
- To Automate the PC shutdowns.
- To regulate the user identity of every PC along with time in a digital format.
- To alert the users about the remaining time of the current lab session
- A proper GUI will be created which will display the important announcements.

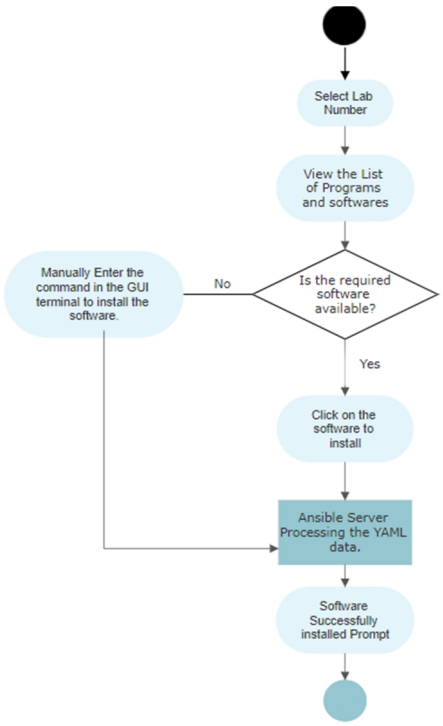

Fig. 3. Proposed System Flowchart

Remote Devices: Desktops and mobiles will be useful for managing the lab which will be further connected to the main web server via internet medium. Here both web based GUI as well as an android application will be developed for more flexibility.

Web Server: Web server shall be used for hosting the whole automated environment. Web server will supervise important characteristics like time zone, host names, user information data and domain Names.

Web Application: The web server will host the required Web application which will be connected to all the lab systems and can be easily managed and controlled via a user-friendly GUI.

Ansible Server: Ansible server will be used for hosting the ansible environment throughout the whole network via the LAN connection and thus all ansible related queries and commands for lab automation can be easily carried throughout.

Client PCs: No matter what operating system the client PC's are running on , any activity can be easily carried out irrespective of what the Operating system is, throughout the network.
As we are using a free open-source platform for our system, many labs can be automated using the same architecture at a very cost-effective valuation.

FTP server: This server will contain all the large files for easy deployment to the local machines in the LAN instead of downloading them from internet individually.

VI. CONCLUSION

The main motive of our work is to create a trustworthy, efficient and real-time system for administration of IT labs in universities. Now all the administrative tasks inside the lab can be executed at a very minimal time and effort with our system. The overall purpose was to minimize the efforts and ensure rapid deliveries of the needed software through automation. These objectives have been checked successfully and we hope to enhance the system furthermore and increase the advancements in our system. Thus we are making an effort to implement this system in the current university labs and modernize the IT labs methodically.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to A.P shah institute of technology for giving us an opportunity to work under this design. We got an opportunity to learn about the practical working of ansible .We would like to thank our college faculties for arranging the necessary apparatus and guiding us throughout the process. We would like to thank our HOD Mr. Kiran Deshpande for assigning us with such an innovative idea to work upon. In the end we would also like to thank our parents for their emotional support and motivation.

REFERENCES

[1] Xavier Decoster and Maarten Balliauw " Automated Delivery in Pro Nuget", October 2016.

[2] D.Palma, December 2016 "Topology and orchestration specification for cloud applications (TOSCA)" , November 2013.

[3] Pavel Mašek Martin and ŠtůsekJan Krejčí, "Unleashing Full Potential of ansible Framework: University Labs Administration", May 2018.

[4] Nishant Kumar Singh , Amity University, "Automated Provisioning of Application" , January 2016.

[5] J.O.Benson, J. J. Prevost, P. Rad, "Survey of automated software deployment for computational and engineering research," in System Conference (Sys Con), 2016 Annual IEEE, pp.1–6, IEEE, 2016.

[6] Jay Shah, Dushyant Dubaria and Prof. John Widhalm "A Survey of DevOps tools for Networking", August 2019.

[7] Wang yiran PetroChina, Zhang tongyang PetroChina and Guo yidong PetroChina "Design and Implementation of Continuous Integration scheme Based on Jenkins and Ansible", May 2018.

[8] Marco Ehrlich, Henning Trsek and Martin Gergeleit "Secure and Flexible Deployment of Industrial Applications inside Cloud-Based Environments", September 2019.

[9] S.A.I.B Suriya Arachchi, Indika Perera "Continuous Integration and Continuous Delivery Pipeline Automation for Agile Software Project Management", June 2018.

[10] Christof Ebert, Gorka Gallardo, Josune Hernantes and Nicolas Serrano "DevOps Tools", April 2016.