Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 34 No. 4 (2024)

Revolutionizing Educational IT: A DevOps-Enabled Cloud Framework for Next-Gen Learning

Bhushan Jadhav[1], Sonali Jadhav[2], Arti Deshpande[3]

[1,] Department of Artificial Intelligence and Data Science

[2,3]Department of Computer Engineering, Thadomal Shahni Engineering College

Thadomal Shahani Engineering College, Bandra West, Mumbai 400050 E-mail address: bhushan.jadhav@thadomal.org

Article History:

Received: *25-08-2024*

Revised: *16-10-2024*

Accepted: *07-12-2024*

Abstract:

In the current scenario, the reliance on information and communication technology (ICT) in education has a number of limitations. Historically, the education sector of managing was extremely labor-intensive and manual, thus creating many a problem regarding the use of educational solutions. The institutes are trying to broaden their scope of functioning with limited infrastructure to meet the demands of the future since education is gaining significant importance.

Emerging technologies such as DevOps and cloud computing can play a prominent role in lessening such adversities and serve the educational sector well. While DevOps refers to different technologies used to automate multiple applications and services-using multiple tools-for continuous integration, continuous testing, continuous deployment, and continuous monitoring, cloud computing is the supply of computing and storage by network-accessible hot-shared resources that match demand. A wide range of commercial cloud systems is extremely compatible with the technologies of DevOps.

This strategy would, therefore, resolve all issues presently encountered by the provision of new educational opportunities and benefits related to cost savings, ease of execution, automated processes, reduced resource usage, etc. The advantages in terms of the benefits for education from the integration of cloud and DevOps solutions are unexcelled in education. This research paper aims to develop a workable automated framework that relieves the education of time-consuming tasks. It discusses the implementation details of the proposed framework using the open-source private cloud platform OpenStack and DevOps tools such as Jenkins and Ansible. The performance characteristics of OpenStack and the DevOPs tools used are also discussed in this paper.

Keywords: performance, characteristics, DevOps, OpenStack, communication

1. Introduction

The introduction of new technology in the education field pretty much changes the whole setup. Any educational institution providing academic, administrative, or any other service depends on human intervention. One such area in need of special attention and development is the variance in infrastructure needs of different student groups that are being catered to by the institution. Each group uses a different IT tool, thus, manual intervention is required to cater to their needs. This sector has continued to use manual techniques that necessitate a higher level of staffing. Most higher education institutions have trailed behind in implementing new IT solutions due to a lack of funds and resources. As a result, several issues arise in terms of processing, storage, and resource utilization. As education becomes increasingly vital, institutes are striving to expand their offerings despite restricted facilities in order to satisfy future demands. To address this issue, this research paper considers the utilization of Cloud Computing and DevOps implementations.

Cloud computing is a network of computing resources that can be located anywhere and shared to address difficulties faced by educational institutions [4]. Integrating an organization's infrastructure and assets with cloud-based software and services expands the organization's options for balancing system management, expense, and security while also improving service quality [6]. Cloud computing can be used as an excellent alternative in colleges and for advanced study. Multipurpose computational infrastructure provides IT departments with more options and flexibility, and it may be reused multiple times. DevOps is a set of methods that integrate software development (Dev) and IT operations (Ops). It seeks to abbreviate the system development life cycle and produce high-quality software on a continual basis. DevOps tools contribute to faster delivery, greater reliability, scalability, and increased collaboration. In this deployment, DevOps in the education sector would be used to automate repeating instructions, reducing the need for human interaction with software. This saves a lot of time because the process is automated, and the resources required are smaller. Less human intervention would mean a lower error rate.

The focus on automation, collaboration, and continuous integration and delivery (CI/CD) makes DevOps an intriguing enhancement to cloud-native concepts, thus fast-tracking the software development life cycle. This paradigm shift in software development and deployment requires a transformation in the paradigm of education so as to provide students with the skills and knowledge set required in this fast-paced landscape[1]. The education industry might employ a combination of accessible open-source cloud technologies, as well as DevOps orchestration and automation tools, to make the process more automated and cost-effective [3]. As a result, this research article presents a paradigm for addressing the current educational restrictions. It offers a cloud-integrated DevOps solution to meet current educational needs, as well as numerous other significant benefits.

2. Literature Survey

All segments of society pay a lot of attention to the growth and access of education. India has sustained initiative towards the growth of the education sector for many years. The problem of creating alternative models of instruction, continuing education, teacher capacity development, and information systems for effective school administration is being approached [5]. The Information and Communication technologies have become more accessible and reliable. This brings the prospect of using ICT for education. The emphasis in traditional education has been on the material of the course. Teachers have taught through lectures and presentations combined with learning exercises to help students consolidate and rehearse the content. Curricula that foster competency and performance are now preferred in modern environments. Curricula are beginning to place a greater focus on skills and on how knowledge can be used rather than on what the information is [6]. As technology becomes more and more involved with current education practices, it calls for more and more manual processes to be automated. Software is one such aspect that constantly needs to be updated in accordance with the changes in curriculum. Different groups of students may require different software’s which, with the limited IT infrastructure may prove to be difficult to provide. Until now, the installation and maintenance of necessary software in schools with insufficient IT resources has been done manually by system administrators. The drawbacks of this approach are it is time-consuming, there are high chances of human error, and with the limited machines available, installation and maintenance of software with the changing educational curriculum is cumbersome [6].

Automation will make it easier for teachers and administrators to deliver their practices quickly and efficiently without the need of additional human intervention or manual interaction. The Internet-based and interactive media are clearly an essential subject of future education and need to be embedded successfully in formal education and learning [7].

The largest platform which provides hosted services over the Internet is Cloud Computing. On demand services that might be private, public or hybrid are provided. Cloud makes it very convenient to access mobiles and computers. Better services are available to the users due to the large-scale computing paths. Because of its many benefits, cloud computing is commonly used in education. Free or low-cost cloud-based platforms are used on a regular basis to facilitate learning systems, knowledge development, study discussions, and social media, which is becoming increasingly important in our lives [8].

Some of the reasons for using cloud in the educational environment are reduced cost, multi-device support, ease of sharing resources, ease of access even remotely, collaboration support, needs only limited infrastructure, personalized education and convenience of access.

DevOps is collaboration between Development and IT Operations to make software production and deployment in an automated & repeatable way. It increases delivery speed of software's and services, thereby increasing the organization's speed. The word 'DevOps' is a combination of two words, 'Development' and 'Operations'. DevOps, likes to divide and conquer, spreading the skill set between the development and operation teams. It also maintains consistent communication. It helps the companies provide better services and performance. DevOps provides better communication and collaboration with the help of alignment of development and IT operations. The lifecycle of DevOps is shown in Fig. 1

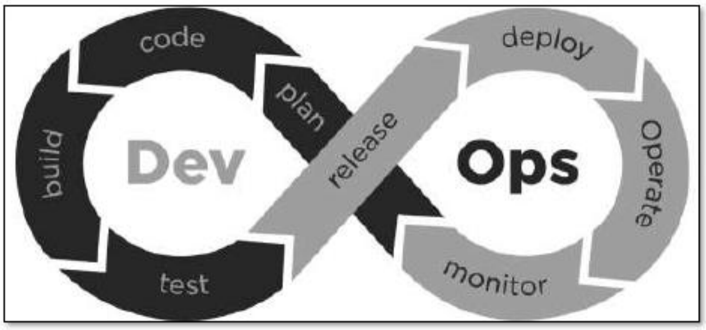


Fig.1. DevOps Life Cycle

The lifecycle of DevOps has eight phases [9].

- Business value and requirements are defined in part by the plan phase. Examples of software for project management and tracking known issues are Jira and Git.
- The code phase comprises both software design and code development.
- During the build phase, you oversee software builds and versions and employ automated tools to assist in the compilation and packaging of code for a later production release.
- Continuous testing either automated or manual is part of the test phase to guarantee the highest possible code quality.
- The release phase entails creating the first software code that the operation team will use.
- The deployment phase may incorporate technologies for scheduling, managing, coordinating, and automating product releases into production.
- Software is managed during production by the Operate phase.
- The monitoring phase entails locating and gathering data regarding problems with a particular software release that is currently in use.

Other than above Lifecycle phases the following are the set of practices are highly used in DevOps. The methodology to implement DevOps in cloud platforms at benefits over traditional IT implementation are defined through the steps mentioned as follows:

a) **Continuous Integration (CI)** requires for the team to be addressed in routine practice where the developer code integration processes which will easily detect the errors. To select the correct tool for CI proves to be a challenge as to needs to have the capability to extend services with different methods in an effective manner,

b) **Continuous Testing (CT)** is one of the most important parts of development. It ensures that the end-user continues to be provided with quality deployments. The DevOps process focuses on the automation of all types of testing as well as appropriate testing environment for the development scenarios. Traditionally, the testing methods involved in the handing away of software from one team to another created a lot of problems. With the use of CT tools, the changes made to the software can be more frequent and error-free. Continuous Testing automates the process of testing.

c) **Continuous Deployment (CD)** is implementation of a pipeline like structure, where the changes occur have to go through a series of state changes and pass all the tests. The quality of these tests ensure that everything is automated and also the quality of the end result while most importantly saving time.

d) **Continuous Delivery** ensures that the software is always presentable and usable. It always needs to be in a finished version.

e) **Continuous Feedback** and monitoring mean the creating of feedback loops and monitoring infrastructure. Thus, bug reports and errors are continuously received to the system which is then solved by taking relevant action.

The need of automation is increasing everywhere. In the education sector software can be used for automation of attendance, assigning tasks, etc. Administrator's manual tasks, installation of the software and its maintenance can also be automated using DevOps tools. The repetitive manual process can be automated with DevOps software in such a way that improvements made on one machine are replicated in all other machines in the network. The challenge for educators is to remove the existing practice of manually carrying out the process and instead focus on the implementation of the proposed solution. Due to the many benefits of DevOps, there is a need for introducing DevOps into the classroom in order to better train students for the new computing industry by helping them grasp realistic DevOps and being knowledgeable of its promising characteristics [10].

3. Proposed System

The proposed framework for education is designed in such a way that it can provide the benefits of cloud computing and DevOps for education. Cloud computing is used for providing the resources (like compute, storage, network) for the applications which are running on the cloud servers inside or outside the premises. It is intended to provide virtual servers to run physical operating system instances along with the applications. The DevOps tools are highly compatible with many Cloud computing platforms which allows to run various DevOps operations. DevOps is used for automation of tasks to be executed on different set of physical or virtual servers. In the proposed system, this task can be scheduled using a DevOps tool called Jenkins, which can run a script for automation.

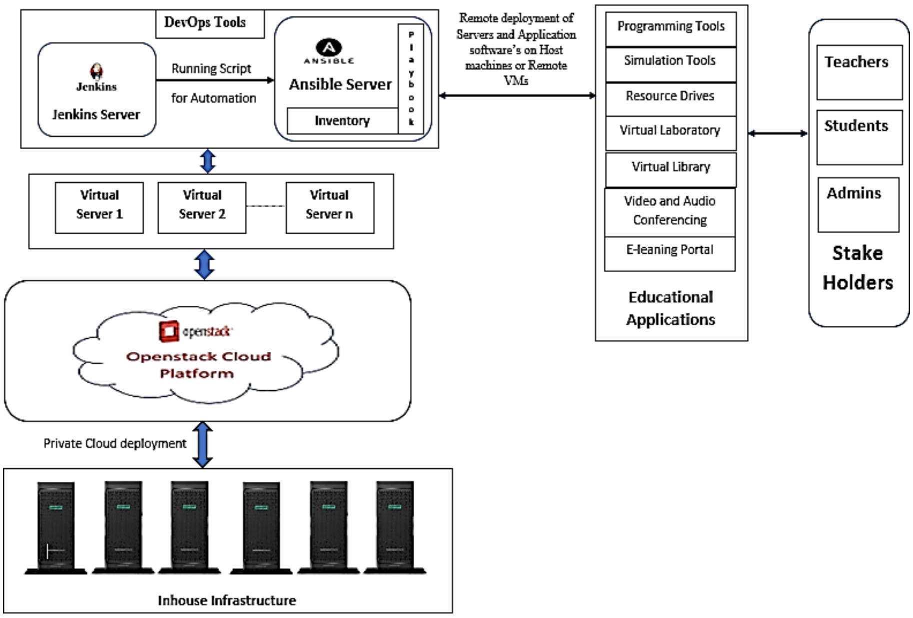


Fig. 2 Proposed framework for education system

This script scheduled by Jenkins runs on another DevOps tool, called Ansible to provision the application and services requested by different stake holders. In proposed framework, the integration of DevOps servers on cloud platform allows to provide eminent features like cost saving, ease of application integration, scalability, high availability, fault tolerance, high speed access, mobility and so on. The proposed framework for education system is shown Fig. 2. The proposed framework composed of following components:

3.1. *In-house infrastructure*

The in-house infrastructure composed of existing hardware like servers, workstations and hosts available inside the institute. Traditionally, these servers need to be managed and monitored using manual operations with the help of cloud computing and DevOps. These servers can be automated for running different automation jobs along with remote monitoring. The compute, storage and the network are the core components of in-house infrastructure, which are provided for deployment of cloud computing platform like OpenStack.

3.2. *Cloud computing platforms*

The proposed framework needs a cloud integration to get high availability, high throughput and faster performance on in-house infrastructure. The private cloud platforms like OpenStack can be deployed within premises to run in-house virtual machines and virtual servers. The OpenStack is an Opensource and highly customized cloud platform with wide variety of components. It allows modifying the functionality of each component as per the requirement. Therefore, it has been selected for implementing the proposed framework because of its uniqueness as compared to other cloud platforms. The Functional architecture of OpenStack cloud platform is shown in Fig.3.

Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 34 No. 4 (2024)

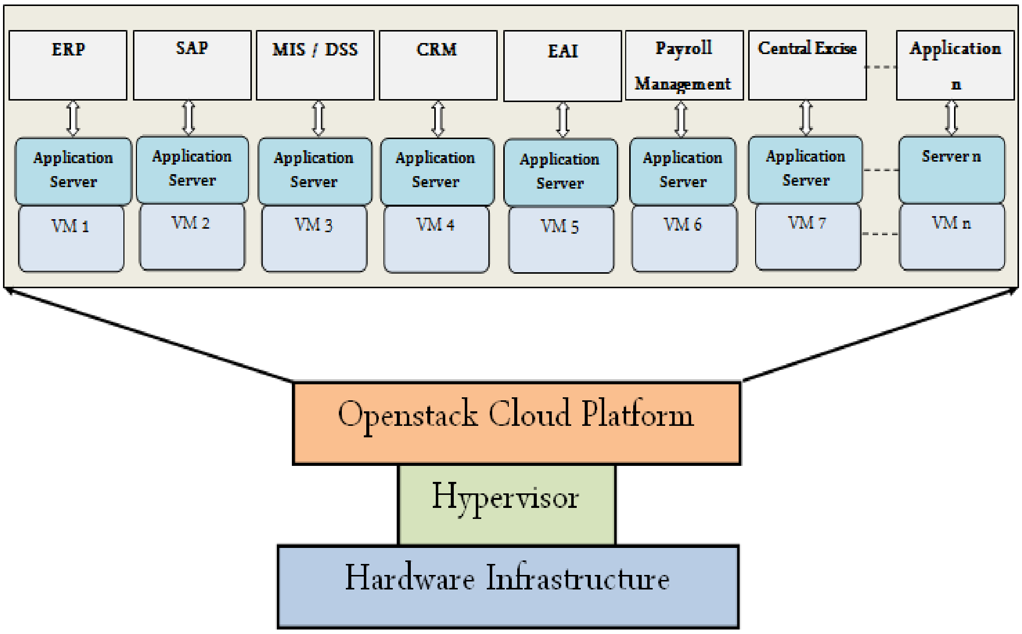


Fig.3 Functional architecture of OpenStack cloud platform

In this setup, the public cloud platforms like AWS can also be used as a secondary cloud to run the services. In case of private cloud failure to sync between private and public cloud, the hybrid cloud integration needs to be implemented by means of using services like virtual private cloud (VPC), load balancing, auto scaling, etc. But for the implementation, the proposed framework uses the OpenStack cloud platforms to deploy and run the different virtual servers, which is to provision the DevOps solutions. It has two core servers like Jenkins and Ansible to be deployed as a virtual machine on hybrid cloud platform.

3.3. DevOps tools

The cloud computing platforms deployed on in-house infrastructure are used to provision the DevOps tools like Jenkins and Ansible. There are many DevOps tools available such as for continuous integration TeamCity, Jenkins, TravisCI, etc. For continuous deployment there are tools like Docker and Kubernetes, while version control is provided by tools like GIT, CVS, etc. The proposed framework mainly uses two DevOps tools like Ansible and Jenkins, which are explained as follows:

3.3.1. Jenkins

Jenkins is an open-source automation tool written in Java with plugins built for Continuous Integration and Continuous Delivery purposes [4] [5]. Jenkins builds and tests the software projects constantly; this makes it simple for the developers to integrate changes to the project. A new build can be easily obtained. The large number of testing and deployment technologies make it possible to deliver software continuously. Jenkins provides machine- consumable remote access API to its functionalities. Jenkins provides a Remote API. A token ID and reference link is provided by the API which helps us to trigger a build. This link can be run locally. Jenkins helps you to trigger a job from a script, command line, or GitHub hook anytime someone commits a move, as well as trigger a job from one server to another. With the help of automation, Jenkins expedites the software development process. Jenkins integrates development life-cycle processes of all kinds, including build, document, test, package, stage, deploy, static analysis, and much more. In proposed system, Jenkins will be used for taking the request form stake holders to provision the requested software application, compile the script and schedule it for deployment over the Ansible server.

3.3.2, Ansible

Configuration management tools can automate the manual task of installing software's. Ansible is one such tool that can be used for the same. It has an added advantage as it is open-source and reliable. It can be used to configure, automate instructions for and manage numerous remote nodes (machines) using a single node (machine). An SSH key is required for gaining access and managing the remote nodes.

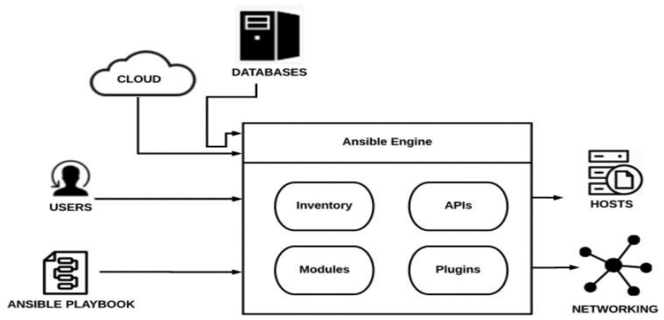


Fig.4 Ansible Architecture

The Ansible architecture is shown in Fig.4. It consists of the following components.

i. Inventories: Ansible consists of a list of hosts (nodes) which are basically the IP addresses of the remote nodes. These can be servers, databases, remote machines etc.

ii. Modules: To execute modules on the remote hosts playbooks are executed. These playbooks can be used to control the system, resources, make API calls to network and install packages.

iii. APIs: APIs are used to access cloud services.

iv. Plugins: Plugins are small code bits that extend the Ansible functionality that are used to perform Ansible tasks. Ansible is provided with several helpful plugins, and your own plugin can be easily created.

Ansible playbooks are used to push the modules to the remote nodes over SSH. The management node is in charge of running the playbook. Using the inventory file, the list of hosts are defined. After establishing SSH connection, the module is run on the management node which runs the playbook on all remote nodes. Thus, Ansible will create a single configuration file containing all commands for the installation of particular applications. In proposed architecture, the ansible server is responsible for delivering the requested software application by the teacher or the stake holders by running a playbook which receives scripted instructions from Jenkins server. For remote deployment of physical or virtual servers and application software on either physical host machine or remote virtual machines.

3.4. Applications

The education softwares composed of repositories for programming tools, simulation tools, virtual laboratory tools, library applications, video and audio- conferencing software and e-learning portals, which are requested by different stake holders to be deployed on cloud or in-house infrastructure using Ansible and Jenkins severs.

3.5. Stakeholders

The stake holders are the users from educational institutes who are intended to run different educational software applications on a daily basis. They are the requester in proposed framework who get access to the application upon executing it on physical or virtual severs scheduled by DevOps tools like Ansible and Jenkins. For example, if a stake holder wants to work on a virtual laboratory, the task will be scheduled on cloud platform to provision a

virtual machine, which is managed by Jenkins server. The Jenkins sever in responsible for executing a script that has instructions to deploy a list of softwares requested by the stake holders. The scripted request will be received by the Ansible server to perform the action using playbook file. Ansible is responsible for automated remote deployment of intended software applications requested by stake holders into a host machine or virtual machine.

4. Implementation Methodology

The implementation of proposed framework involves two stages i.e., deployment of OpenStack cloud platform and deployment of DevOps tools like Jenkins and Ansible.

4.1. Deployment of OpenStack cloud platform

The deployment of OpenStack cloud platform has performed on DELL Studio Workstation Server along with Ubuntu server version 18.04, 64-bit distribution for the deployment of OpenStack Mitaka release with the help of MaaS and juju. Metal as a Service (MaaS) is used for data center management (DCM) by automating and optimized provisioning for production hardware. MAAS can perform zero-touch deployments of Windows, Ubuntu, CentOS, RHEL and SUSE. MaaS in combination with Juju, can easily model and deploy complex environments. It can pull down and redeploy instances easily which are entirely abstracted from underlying infrastructure. The MaaS can be installed in one of the servers connected to internal and external network. The following command is used for installing MaaS over Ubuntu Server.

⇨ $ sudo apt-get install maas maas-dns maas-dhcp maas-proxy

Once Maas is installed it can be accessed using the IP address of the ubuntu server. The physical nodes connected in the same network of MaaS server are automatically detected and commissioned. The juju can be installed using following command

⇨ $ sudo apt-get install juju-core

Once MaaS and juju is deployed the OpenStack and other services can be deployed over juju server using following command which does the automated provisioning of required virtual server.

⇨ $ sudo juju deploy <Server_name>

The command for installing OpenStack on juju would be

⇨ $juju deploy OpenStack

To install Jenkins and Ansible on OpenStack, following commands would be used

⇨ $juju deploy ansible

⇨ $juju deploy Jenkins

Once required components of Juju are installed with their relationship. The juju management portal can be accessed using the IP address of juju controller . The juju deployment dashboard is depicted in Fig.5

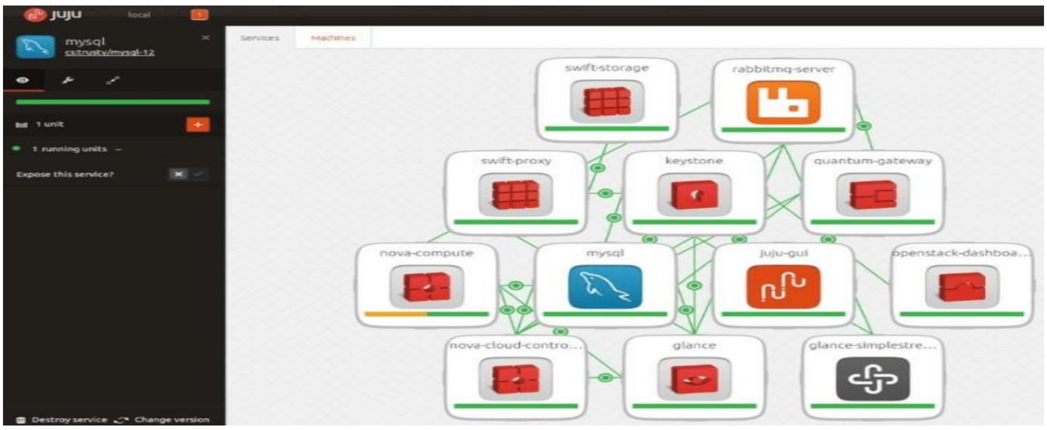


Fig. 5 Juju deployment dashboard

Once deployment is over the OpenStack dashboard can be accessed using IP address of compute server.

4.2. Deployment and configuration of DevOps tools

The steps to get deploy and configure Ansible are as follows [5]:

Step1: First we establish connection with the target nodes using SSH. The commands for doing this are as follows

$: ssh-keygen

$: ssh-copy-id root@[ip of node]

Step 2: IP address of target machines are added to the the hosts file so that ansible knows who to connect with. The IP addresses of the remote nodes are added to a group name.

vim /etc/ansible/hosts [testnodes]

192.168.1.117 ansible_user=root

192.168.1.121 ansible_user=root

Hence both IPs can be called using name testnodes

Step 3: The inventory file is modified to contain the SSH details of target machines. The commands for doing it would be as follows:

$: mkdir test-project

$: cd test-project

$: cat > inventory.txt Type in:

$: target1 ansible_host=192.168.1.117 ansible_ssh_pass=OpenStack@123

$: target2 ansible_host=192.168.1.121 ansible_ssh_pass= OpenStack@123 Hence, remote host IP addresses and password are added.

Step 4: We create a YML file for writing the playbook Make a directory playbook inside /etc/ansible.

And create a yml file for writing play for installing For eg. Apache on nodes.

$: mkdir /etc/ansible/playbook

$: vim /etc/ansible/playbook/pythoninstall.yml

$: vim /etc/ansible/playbook/javainstall.yml

Step 5: We use the following command to run the playbook

$: ansible-playbook javainstall.yml

The Steps to configure and run a Jenkins jobs is given as follows:

Step 1: Create a new Jenkins job and make a dynamic inventory with the following instructions:

=> Click on “New Item”

=> Select the option “This project is parameterized”

=> Create a multi-line string parameter called “$INVENTORY”

=> Select “Invoke an ansible playbook” (After Ansible plugin is added this option will become visible)

=> Enter the path to yml file in local storage

=> Add User ID and Password details of destination nodes

=> Save

Step 2: Run the created Jenkins file

=> Open the Jenkins job

=> Click on “Build with parameters”

=> Enter the list of remote target nodes IP addresses

=> Build

Go to latest build

=> Click on “Console output” to the see the output

5. Results and Discussion

The proposed framework has been implemented with the help of Cloud Computing platforms and DevOps tools in the last section. This section proposes the results which are found during the testing of the framework. In the first phase, the private cloud platform like OpenStack has been implemented followed by integration of DevOps tools in it. The result of OpenStack Dashboard is shown in Fig.6.

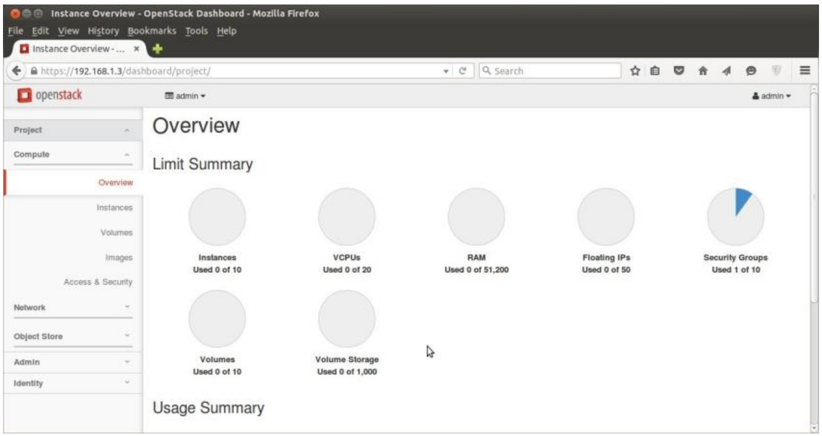


Fig. 6 OpenStack dashboard

To check the performance of OpenStack cloud platform, the benchmark is calculated on inhouse physical server and VM over the OpenStack server. For testing the performance, the hardware configuration used for both are kept same. The comparison betweena dedicated standalone server and proposed system based on parameters like response time and average speed are shown in Fig. 7.

The response time of the proposed system is calculated by recording the time taken by standalone machine and OpenStack VM to perform read operation with different file size and the average speed is calculated by counting the IOPS by transferring the file between standalone system and OpenStack based cloud Virtual machine.

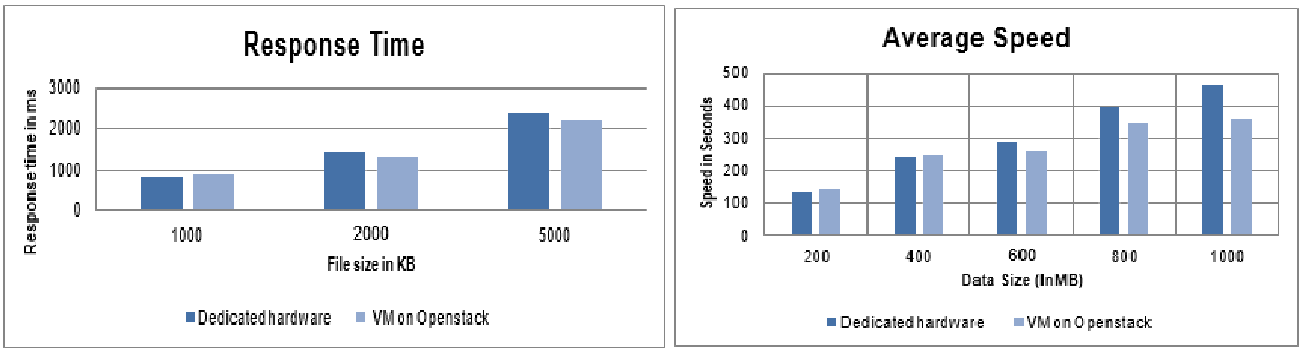


Fig. 7 Response time and Average Speed in OpenStack Cloud Platform

From above Figure it is seen that the Virtual machine running over OpenStack gives better response time in performing read operation over different file sizes and average speed for performing IO operations over

OpenStack cloud is performing well in all the aspects than dedicated hardware.

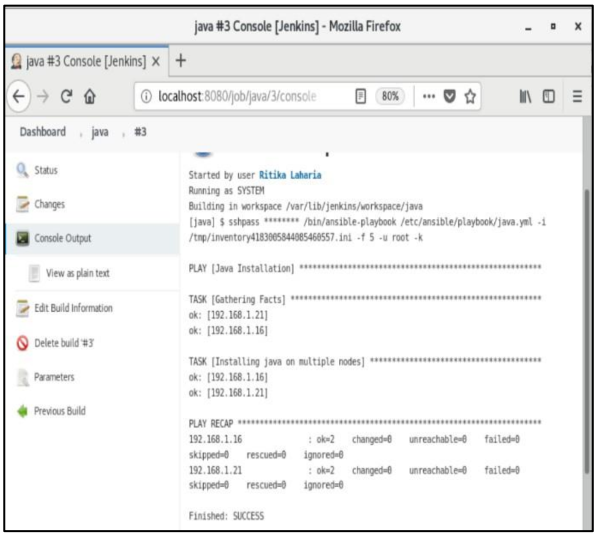
The Second phase involves testing the various deployments of software applications on DevOps tools like Jenkins and Ansible requested by various stakeholders. The results for implementing the proposed framework are based on automation jobs written inside Ansible and parameterized by the Jenkins server. Two playbooks have been developed for installing python and java into the hosts defined in the inventory file during the implementation stage. Fig. 8 shows the Playbook script for Java and python. The Script must be deployed by Jenkins's server to make the successful deployment of required software's by the students.

```yaml
---
- name: Java Installation
  hosts: all
  tasks:
  - name: Installing java on multiple nodes
    yum:
        name:
          - java-11-openjdk-devel
        state: present
```

```yaml
---
- name: Python Installation
  hosts: all
  tasks:
  - name: Installing python on multiple nodes
    yum:
        name:
          - python3
        state: present
```

Fig. 8 Playbook Script for Java and Python

Fig. 9 shows the Execution results for deploying Java and Python in host machines by using yaml script on Jenkins Server. This script deploys application software's like Java and Python in all the hosts specified in the inventory file. For this experiment, only two hosts have been added into inventory with IP address 192.168.1.16 and 192.168.1.21



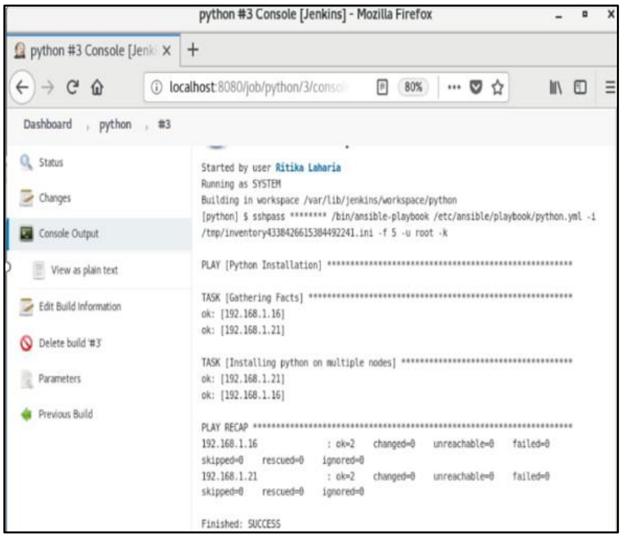


Fig. 9 Execution results for deploying Java and Python in host machines

Similarly, the administrator could install, on the host system or VM allocated to teachers or the students, any software that teachers or students have requested. The Jenkins Management uses the web portal to send the job to be executed. From the portal, an administrator could prepare a yaml script that would run on the Ansible server to install or deploy certain applications or servers on the host system. This research paper also examines and compares the efficacy of general configuration management software.

For this, the experiment was intended to be done on three different deployment platforms: Puppet, Chef, and Ansible. The same hardware infrastructure was maintained for all three platforms. The time taken by three platforms like Puppet, Chef, and Ansible, to deploy the same set of tasks has been recorded. The graph has shown the execution time taken by Puppet, Chef, and Ansible.

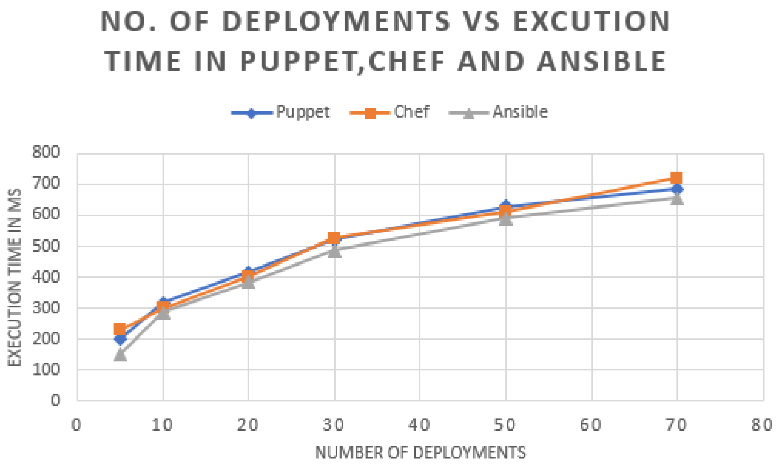


Fig. 10 Results of execution time taken by Puppet, Chef, and Ansible

From above Fig.10 , it can be seen that Ansible gives better performance for deployments as compared to Chef and Puppet in variable deployment size. However, Chef and puppet gives slower performance than Ansible when number of deployment increases. This implies that it would be a better solution for existing educational setup where task to be performed are different in types and numbers. The Ansible is highly suitable for such cases. Therefore, it is found to be a total solution for educational needs [5].

Jenkins and Ansible are included together in this research paper. When the prescribed mixture is used, the experiment yields better results. Other DevOps software, such as TravisCI, CircleCI for continuous integration and Puppet, Chef for configuration management, may be used for this experiment. Table I shows the comparison between Continuous Integration (CI) tools, while Table II shows a comparison between Configuration Management tools.

Table 1 - Comparison between Continuous Integration (CI) tools

Jenkins	TravisCI	CircleCI
ıpported on Linux, MacOS and Windows	is supported on Linux and Windows	CircleCI is supported on Linux and MacOS
Jenkins is open-source software	TravisCI is not open-source software	CircleCI is not open-source software
Jenkins can be set up locally or in the cloud	TravisCI is set up in the cloud	CircleCI is set up in the cloud
opensource and free of cost software	Cost of TravisCI is relatively higher	Cost of CircleCI is relatively higher
ʻides built-in analytics through plugins	ɔrovides analytics through a web-based platform	Analytics in CircleCI is provided by default
Jenkins supports a large number of plugins	CI supports relatively a smaller number of plugins	ıports relatively a smaller number of plugins
Jenkins supports all types of containers	ɜupports a limited number of containers	ɜupports a limited number of containers

Table 2 - Comparison between Configuration Management tools

Ansible	Puppet	Chef
Ansible is relatively easy to setup	Puppet is complex to setup	Chef is complex to setup
: is a push-based Configuration Management tool	t is a pull-based configuration management tool	s a push-based configuration management tool
:cture of Ansible consists of only Master (Agentless)	re of Puppet consists of Master with agent	of Chef consists of Master with agent
Ansible can be configured in Python, YAML	t can be configured in Ruby, Embedded Ruby	Chef can be configured in Ruby
Cost of Ansible is relatively less	Cost of Puppet is relatively high	Cost of Chef is relatively high

6. Conclusion

Automation in IT advances basic as well as complicated tasks for all systems, across all types of applications. Thus, IT automation assumes a significant role in human intervening industries like education. Nowadays, the education system faces many problems, which lead to collaboration issues, poor resource organization, and limited active infrastructures. This paper attempts to provide a framework for a cloud-integrated DevOps approach using OpenStack as a cloud platform, Jenkins, and Ansible as a DevOps tool. The proposed framework seeks to rectify the shortcomings of the present education sector and ease the long manual processes. Thus, based on our extensive study, we can conclude that the proposed system, based on cloud computing and DevOps, is simple and allows the untroubled and automated installation of applications into hosts and servers using infrastructure as code, easy access to resources, faster deployment of applications whose availability would be guaranteed with high fault tolerance and performance, etc. Some disadvantages of the suggested framework are less processing power needed for running OpenStack cloud and integration of cloud DevOps being more complex, which can be solved using containerization tools, like Docker or Kubernetes after some time.

References

[1] P. Mane, J. Kallimani, B., and L. Kale, "Innovative Pedagogical Approaches for Teaching Cloud-Native DevOps: Integrating Theory and Practice," *Nanotechnology Perceptions*, 2024. doi: 10.62441/nano-ntp.v20.

[2] S. Mathew, "Implementation of Cloud Computing in Education – A Revolution," *International Journal of Computer Theory and Engineering*, vol. 3, pp. 473-475, June 2012.

[3] G. Deka and M. Borah, "Role of Cloud Computing in Education," *2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*, 2012. doi: 10.1109/ICCEET.2012.6203884.

[4] T. Noncheva, "Learning Without Limits: Cloud Technologies Are Expanding Educational Horizons," *Education and Technologies Journal*, 2024. doi: 10.26883/2010.241.5983.

[5] K. Hasija, R. Laharia, M. S. Apurva, and B. Jadhav, "Novel Approach for Implementing DevOps for Education," *International Journal of Computer Applications*, vol. 174, no. 31, pp. 46-51, Apr. 2021.

[6] "National Policy on Information and Communication Technology (ICT) in School Education."

[7] S. Amin, "An Effective Use of ICT for Education and Learning by Drawing on Worldwide Knowledge, Research, and Experience: ICT as a Change Agent for Education (A Literature Review)," *Scholarly Journal of Education*, vol. 2, no. 4, pp. 38-45, Apr. 2013.

[8] H. M. Wasfy et al. and T. M. Wasfy, "The Education Sector Revolution: The Automation of Education,"

ASEE Annual Conference and Exhibition, American Society for Engineering Education, June 2013, pp. 8044.

[9] E. A. Ahmed and H. A. Ahmed, "A Proposed Model for Education System Using Cloud Computing," *2018 3rd International Conference on Emerging Trends in Engineering, Sciences and Technology (ICEEST)*, 2018, pp. 1-4. doi: 10.1109/ICEEST.2018.8643331.

[10] Akbar, M., Rafi, S., Alsanad, A., Qadri, S., Alsanad, A., & Alothaim, A. (2022). Toward Successful DevOps: A Decision-Making Framework. IEEE Access, 10, 51343-51362. https://doi.org/10.1109/ACCESS.2022.3174094.

[11] P. Agrawal and N. Rawat, "DevOps, A New Approach to Cloud Development & Testing," *2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT)*, 2019, pp. 1-4. doi: 10.1109/ICICT46931.2019.8977662.

[12] G. Rong, S. Gu, H. Zhang, and D. Shao, "DevOpsEnvy: An Education Support System for DevOps," *2017 IEEE 30th Conference on Software Engineering Education and Training (CSEE&T)*, 2017, pp. 37-46. doi: 10.1109/CSEET.2017.17.