

# Novel Approach for Implementing DevOps for Education

Khushbu Hasija  
Thadomal Shahani  
Engineering College  
Mumbai 400050

Ritika Laharia  
Thadomal Shahani  
Engineering College  
Mumbai 400050

M.S. Apurva  
Thadomal Shahani  
Engineering College  
Mumbai 400050

Bhushan Jadhav, PhD  
Thadomal Shahani  
Engineering College  
Mumbai 400050

## ABSTRACT

In the current scenario, the use of digital technologies in education are limited. Due to the limited IT infrastructure, most of the schools and colleges are lagging behind to acquire the new IT solutions. This limitation forces them to reuse their infrastructure for multiple activities. Most of the activities happening in the education field are manual with involvement of extra manpower to manage it. With these limitations, every year different students need different softwares to be operated for implementing the practical sessions. They have to constantly be updated to the latest version of softwares or retain a particular version that the teacher or curriculum requires. The system administrators have to manually configure softwares on each and every machine present which is cumbersome. The objective of this research work is to propose a system that can make this tedious process automated where the changes made in one machine is reflected in all the other machines present in the network with the help of DevOps tools. This approach would overcome all the existing challenges by providing new opportunities in the education sector. The IT automation provided by DevOps is the use of software to make repeatable instructions and processes to exchange or reduce human interaction with IT systems. The advantage of this approach is that; it is time saving, can be deployed over lesser resources and the entire system would be automated.

## Keywords

DevOps, CI/CD, Jenkins, Ansible, Infrastructure as Code

## 1. INTRODUCTION

The modern world is the age of information technology. Technological advancements have impacted all sectors of life including communication, health, transportation and education. A sector which has not seen much automation of services is the education sector. In the current scenario, the use of digital technologies in education are limited. Due to the limited IT infrastructure, most of the schools and colleges are lagging behind to acquire the new IT solutions. This limitation forces them to reuse their infrastructure for multiple activities. While these limitations exist, every year different students need different software to be operated for implementing the practical sessions. The machines have to constantly be updated to the latest version of software or retain a particular version that the teacher or curriculum requires. A common issue of this is that the system administrators have to manually configure software on each and every machine present which is cumbersome.

DevOps is one of the technology which can be used in the education sector. DevOps tools consists of collaboration tools, planning tools, developer environment tools, continuous integration tools, and deployment tools [1]. The IT automation provided by DevOps is the use of software to make repeatable instructions and processes to exchange or reduce human

interaction with IT systems. The advantage of this approach is that; it is time saving, can be deployed over lesser resources and the entire system would be automated. Thus, DevOps approach has huge capability to overcome the challenges of the existing scenario of the education sector.

In this research paper, the approach for introducing DevOps in the education sector is discussed along with the implementation. The proposed model uses a combination of available open-source DevOps orchestration and automation tools to help provide a system for the education sector where everything would be automated and cost-effective.

## 2. LITERATURE SURVEY

The literature related to existing education system and current practices in DevOps is explained in further subsections.

### 2.1 Existing Practices in Education

In corporate and academic circles, education is important. Training or preparation in the first place is used to make workers do something better. Last but not least, curriculum succeeds at generating interest in students' minds. The use of technologies will in any situation allow students to better understand and save principles. The importance of technology in education is quadruple: it is used in the curriculum, as a teaching device, as a platform to help and enrich the whole process of learning. Education is passive, reactive and violent, thanks to the technology [8]. A challenging area in the field of education is the consistent rise in need for digitization and automation of resources. Majority of the activities happening in the education field are manual with involvement of extra manpower to manage it. IT resources in educational institutions are often reused for students of different departments. To date, the process of installation and maintenance of required software in schools with limited IT resources has been manual and carried out by system administrators. The major disadvantages of this process are:

- a) Manually carrying out the same set of commands on a fleet of machines is time-consuming.
- b) Chances of human error being present are high.
- c) With the limited machines available, constant installation and maintenance of software in accordance with the changing educational curriculum is cumbersome.

As a result of these drawbacks, this process in the education sector is increasingly in need of automation. 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 [9].

### 2.2 Current DevOps Practices

Julio Sandobalin et al. in [4] have introduced a proposal for an end-to-end automation of infrastructure provisioning in the Cloud through a toolchain using DevOps community tools.

They have come up with an effective automation of the Infrastructure as Code concept and how it can be implemented with the DevOps community tools. In [5], Eduard Luchian, et al. propose a solution that brings together three very powerful software packages in order to provide a true cloud deployment, i.e. Chef, Vagrant and VirtualBox. This type of automatic setup has the possibility to easily keep up with the new OpenStack releases. Thus it needs just a few line changes to upgrade the installation to a newer version. The work proposed by Nishant Kumar Singh, et al. in [6] presented a new way of deploying applications on cloud using automated provisioning. It shows how flexibility of Ansible makes it easy to automate the whole procedure of deployment of applications. The research done by Gideon Juve and Ewa Deelman in [7], presents the design and implementation of a system used for automatically deploying distributed applications on infrastructure clouds. The system interfaces with several different cloud resource providers to provision virtual machines, coordinates the configuration and initiation of services to support distributed applications, and monitors applications over time. After reviewing the following papers it can be concluded that using Ansible as the DevOps community tool for the small scale application in the education sector is because of its flexibility and the fact that it is open source.

### 3. PROPOSED SYSTEM

The current practices of using digitized solutions for education are incapable of solving the limitations in the existing environment. Therefore, the introduction of automation technologies provided by DevOps could be a welcome option in education. The methodology for using DevOps in education is explained in subsequent sections.

#### 3.1 Automation in Education using DevOps

DevOps approach can be an option for automating the system administrator's manual tasks. With the help of DevOps tools, the tedious manual process can be automated in a way where the changes made in one machine are reflected in all the other machines present in the network. As DevOps culture continues to rise with the growing market, there is also a subsequent rise in the number of DevOps tools available. They consist of collaboration tools, planning tools, monitoring tools, configuration management tools, continuous integration tools and deployment tools. The following section discusses the options within DevOps tools which can help automation of manual processes and provide significant improvements in the processes within the education sector.

- a) Collaboration tools: Collaboration tools are important for any team to work together. Collaboration tools help teams to achieve their common goal of communicating with each other to log performance and solve problems. Common collaboration tools include Slack, a collaborative communication tool created for business communication, and JIRA, a project-management tool used for planning and working together on a software.
- b) Planning tools: Planning tools enforce DevOps values such as coordination, openness, and teamwork. It can be used to exchange data, assign task ownership, maintain accountability, and much more [1]. JIRA along with being used for collaboration can also be used for planning as it is an assignment-tracking tool. It provides Kanban-

style boards that visualise the progress of tasks, who the task is from and recent developments in it.

- c) Monitoring tools: DevOps monitoring tools help organisations monitor their systems to improve visibility and productivity which help reduce downtime, allocate resources, monitor network storage compute etc.
- d) Configuration Management tools: Configuration Management tools automate repetitive, manual and error-prone tasks, it drives automation to produce consistency in systems. DevOps, abstracts the environment like a configuration file which can be deployed easily and any changes are made systematically rather than manually [1]. Ansible is one such open-source configuration management tool used for IT automation, it is written in python.
- e) Continuous Integration tools: Continuous Integration tools allow users to interact and merge their work into a singular repository. Jenkins is one of the popular continuous integration tool that can be used for automated building, testing, and integration of changes.

### 3.2 Ansible for Education

Installation of software currently carried out manually by system administrators is a process that can be automated using configuration management tools. For configuration management, Ansible is a simple, dependable, and reliable tool.

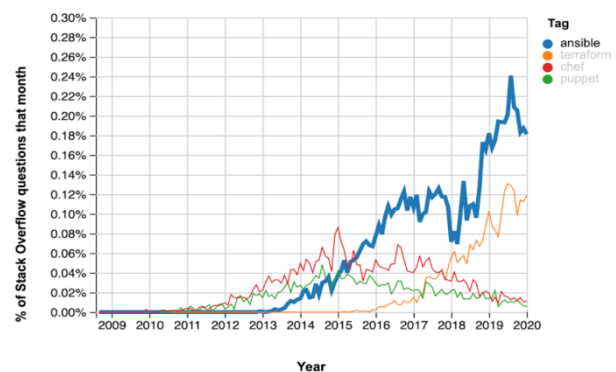
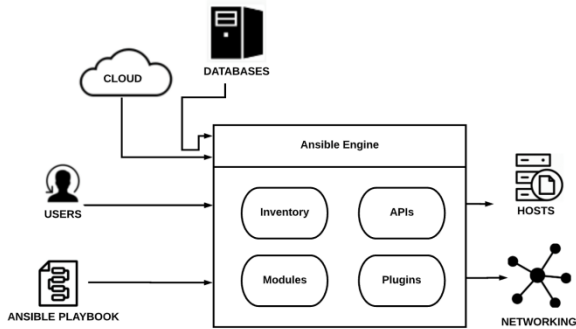


Fig.1 Performance Ansible related to other technologies

Ansible requires a password or an SSH key for managing systems. The performance of Ansible related to other technologies like Terraform, Chef, and Puppet, as can be seen in the graph below Fig.1 [3].

Deploying using the ansible crowns top in its group as it is an open-source, efficient, powerful, easy to setup and agentless.



**Fig.2 Ansible Architecture**

The Ansible architecture is shown in Fig.2. It consists of the following components.

- **Inventories:** Ansible inventories are lists of hosts (nodes) that need to be handled, including their IP addresses, servers, databases, and so on. Ansible takes action using a transport – SSH for UNIX, Linux, or Networking devices, and WinRM for Windows systems.
- **APIs:** In Ansible, APIs are used to transport public and private Cloud services.
- **Modules:** Playbooks are used to execute modules directly on remote hosts. The modules are capable control system resources. System files, making API calls to the service network and installing packages are acted upon by the modules.
- **Plugins:** Plugins are small pieces of code that expand the functionality of Ansible and are used to run Ansible tasks. Ansible comes with many useful plugins, and one can easily build their own plugin.

Ansible pushes small programs to the connected nodes which are called the Ansible modules over SSH. . Any computer can access the module library. The machine on which the installation is done is the management node which is in charge of the playbook's entire execution. The inventory file consists of the list of hosts which can execute the Ansible modules. After an SSH connection is established he modules are run on the host computer which installs the software. In this way Ansible is capable of creating a single configuration file that contains all the commands to install specific software.

### 3.3 Jenkins for Education

Jenkins is an open-source automation tool [2] written in Java with plugins built for Continuous Integration and Continuous Delivery purposes. Jenkins makes it easier to integrate changes to the project for developers by building and testing the software projects continuously. It also makes it simpler to obtain a new build. Software can be continuously delivered by integrating it with a large number of testing and deployment technologies. Jenkins provides machine-consumable remote access API to its functionalities. Remote access API is offered in a REST-like style. That is, there is no single entry point for all features, and instead they are available under the ".../api/" URL where "... " portion is the data that it acts on [2]. The software Jenkins is an open-source CI/CD tool that is capable of orchestrating a string of actions that achieve Continuous Integration.

### 3.4 Continuous Integration / Continuous Delivery (CI/CD)

Continuous Integration is an approach where the code is updated in a central server by all participants.. Continuous Delivery incorporates the practice of automating the whole product release process on top of Continuous Integration. These procedures are fully automated, fully logged and accessible to the entire team with each run. The stages of CI/CD pipeline is shown in Fig.3.

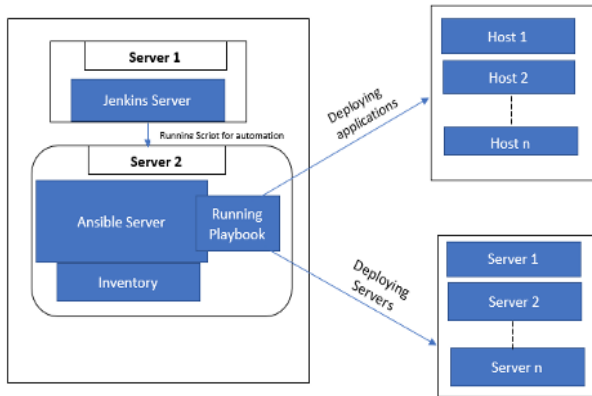


**Fig.3 Stages of CI/CD pipeline**

- **Source Stage:** The source code server triggers the execution. The tool used in this stage gives an alert whenever changes to the code is made by any contributor.
- **Build Stage:** The source code and its dependencies are combined to create a running instance of the software that can be delivered to the end user. If one fails to get through the construction process it can lead to issues in configuration of the project
- **Test Stage:** This step runs automated tests to verify the correctness of the code and the product's actions. The test stage functions as a safety net, preventing end-users from being exposed to easily reproducible bugs.
- **Deploy Stage:** Once the instance passes the testing stage it is deployed. This is the last stage where the software is deployed onto the server once the build passes through all the stages.

### 3.5 Proposed System

The proposed system is based on two DevOps tools called Ansible and Jenkins which are used to do automation of software packages in all the hosts. The proposed system shown in Fig.4 would be deployed onto two physical servers, the first server is used for executing the CI/CD automation script for deploying the resources onto the servers and hosts with the help of Ansible server. The Ansible server is responsible for creating the inventory of all the hosts and servers involved in the current network setup. The Ansible playbook is a yaml file running on the Ansible server called using Jenkins to automatically deploy the required applications onto the hosts or the group of servers like web servers, database servers, application servers, etc.



**Fig.4 Proposed system**

Upon implementing the proposed model, the education system would save the infrastructure by means of running hosts and servers as a virtual machine over the cloud platforms like AWS or Azure. This system will provide the unique benefits like automation of various deployments, quick deployment of required applications, easy accessibility of applications, avoid repetition of installation and so on.

#### 4. IMPLEMENTATION METHODOLOGY

The implementation of the proposed system involves installing the two servers with Jenkins and Ansible. The Ansible server should be configured based on the number of hosts and servers to be connected using an inventory file. The steps for configuring Ansible is given as follows:

In Ansible the management node controls the entire execution of the playbook. Installations are run from this node. The inventory file provides the list of hosts where the Ansible modules needs to be run and the management node does a SSH connection and executes the small modules on the hosts machine and installs the product/software. The module is removed once the module is executed

The steps for configuring Ansible are as follows:

**Step1: ESTABLISHING CONNECTION WITH TARGET MACHINES USING SSH:**

```
# First create ssh key-gen in ansible controller
$: ssh-keygen
# And copy the key to the nodes
$: ssh-copy-id root@[ip of node]
```

**Step 2: ADDING IP ADDRESSES OF TARGET MACHINES TO HOSTS FILE OF ANSIBLE**

This is so that ansible knows who to connect to. Then the IP addresses of remote hosts are added under a group name  
# vim /etc/ansible/hosts  
Type in this: (use ifconfig and find IPs of your target vms)  
[testnodes]  
192.168.1.117 ansible\_user=root  
192.168.1.121 ansible\_user=root  
Note: Here both IPs can be called using name testnodes

**Step 3: CHANGE THE INVENTORY FILE AND ADD THE TARGET MACHINES TO IT**

```
$: mkdir test-project
```

```
$: cd test-project
$: cat > inventory.txt
Type in:
$: target1 ansible_host=192.168.1.117
ansible_ssh_pass=osboxes.org
$: target2 ansible_host=192.168.1.121
ansible_ssh_pass=osboxes.org
Here, remote host IP addresses and password are added.
```

**Step 4: CREATE A YML FILE FOR WRITING PLAY**

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/lampinstall.yml
$: vim /etc/ansible/playbook/pythoninstall.yml
$: vim /etc/ansible/playbook/javainstall.yml
```

The playbook for installing Apache, Python and Java is shown in Fig.5

```
---
- name: Apache Installation
  hosts: all
  tasks:
  - name: Installing lamp on multiple nodes
    yum:
      name:
        - httpd
        - php
      lock_timeout: 180
      state: present
  - name: Start the services
    service:
      name: httpd
      state: started
  ...

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

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

**Fig.5 Playbook for installing Apache, Python and Java**

**Step 5: RUN THE PLAYBOOK USING THE BELOW COMMAND**

```
$: ansible-playbook lampinstall.yml
```

Ansible yaml files can be run using a Jenkins job. An in-built Ansible plugin is available in Jenkins. Using the plugin, Ansible playbooks or ad hoc commands in the build step. During the build, the IP addresses of the destination machines is passed as multi-line string parameter. These parameters act as a dynamic inventory. Different destination IPs can be used to build Jenkins jobs. A Remote API provided by Jenkins, helps in triggering a build with the help of a token ID and a

reference link, which can be run locally. Using these links the website created, will run these links and complete the project.

The steps for running the inventory and playbook using Jenkins connected to Ansible are as follows:

**Step 1: MAKE A NEW JENKINS JOB:**

- Click on “New Item”
- Select a “This project is parameterized”
- Create a multi-line string parameter called “\$INVENTORY”  
(This will act as a dynamic inventory)
- Select “Invoke an ansible playbook”
- Enter the path to yml file in local storage
- Add SSH details of destination machines. (User ID and Password)
- Click on “Save”

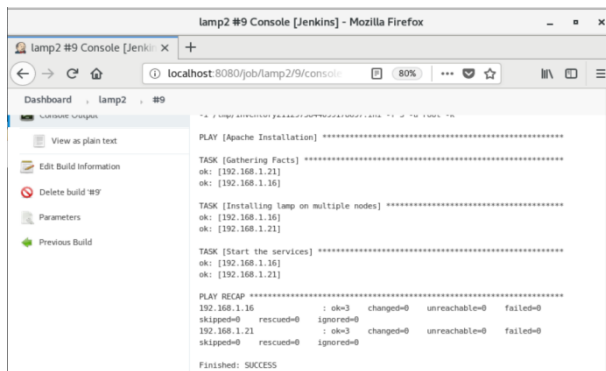
**Step 2: RUN JENKINS JOB**

- => Open Jenkins job
- => Click on “Build with parameters”
- => Enter the list of remote destination IP addresses
- => Click “Build”
- Go to latest build
- => Click on “Console output” to see the deployment

**5. IMPLEMENTATION RESULTS**

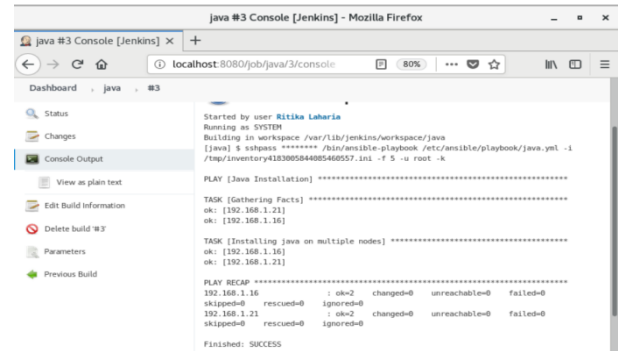
The results for implementing Ansible and Jenkins are based on automation jobs which are written inside Ansible playbook and parameterised through the Jenkins server. In the implementation stage three playbooks have been created for installing python, java and a LAMP server into the hosts specified in the inventory file.

Fig.6 shows the successful execution of yaml script to install LAMP in all the hosts specified in the inventory file. For this implementation it will get installed onto two hosts with IP address 192.168.1.16 and 192.168.1.21



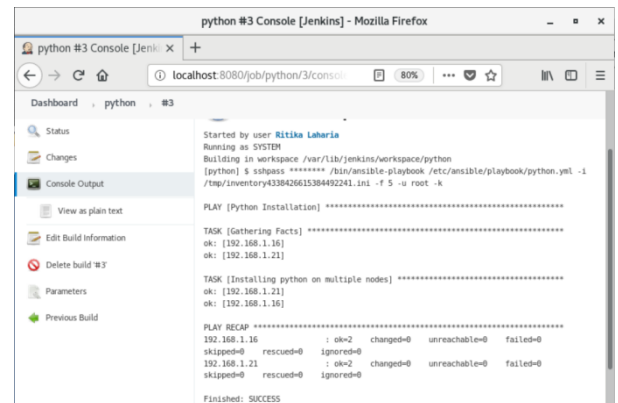
**Fig.6 Successful execution of yaml script to install LAMP in all destination machines**

Fig.7 shows the successful execution of yaml script to install Java in all the hosts specified in the inventory file. For this implementation it will get installed onto two hosts with IP address 192.168.1.16 and 192.168.1.21



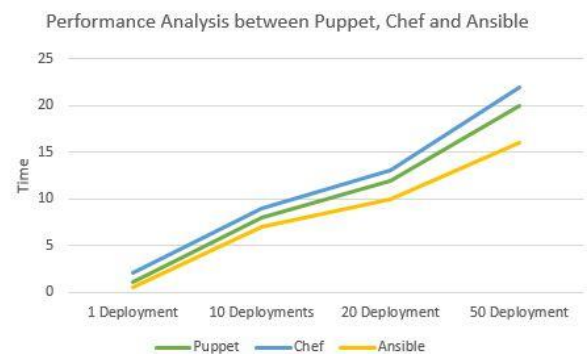
**Fig.7 Successful execution of yaml script to install Java in all destination machines**

Fig.8 shows the successful execution of yaml script to install Python in all the hosts specified in the inventory file. For this implementation it will get installed onto two hosts with IP address 192.168.1.16 and 192.168.1.21



**Fig.8 Successful execution of yaml script to install Python in all destination machines**

In this research paper, the performance between popular configuration management tools are also tested and experimented. The experiment have done using the different deployment of tasks on three platforms like Puppet, Chef and Ansible. All three platforms are deployed on similar hardware infrastructure. The time taken to deploy different set of tasks are recorded. Fig.9 shows the performance analysis between Puppet, Chef and Ansible.



**Fig.9 Performance Analysis between Puppet, Chef and Ansible**

From above Fig.9, it can be seen that Ansible gives better performance for deployments as compared to Chef and Puppet. 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.

### 5.1 Comparative Analysis

In this research paper the combination of Jenkins and Ansible is used. The experiment shows the better results when the given combination is used. However this experiment could be performed using other DevOps tools like TravisCI, CircleCI for continuous integration and Puppet, Chef for Configuration Management. The comparison between Continuous Integration (CI) tool is given in Table 1 and the comparison between software Configuration Management tools is given in Table 2.

**Table 1: Comparison between CI tools**

	Jenkins	TravisCI	CircleCI
Platforms	Linux, MacOS, Windows	Linux, Windows	Linux, MacOS
Opensource	Yes	No	No
Setup	Local and Cloud based	Cloud based	Cloud based
Performance	High	Moderate to High	High
Cloud support	Opensource and proprietary	Proprietary	Proprietary
Parallelism	High	High	Moderate
Cost	Free	High	High
Analytics	Available with plugins	Available in web based platform	Available by default
Plugins support	Rich	Moderate	Moderate
Container support	All types of containers	Limited	Limited

**Table 2: Comparison between Configuration Management tools**

	Ansible	Puppet	Chef
Setup	Easy	Complex	Complex
Configuration Management	Pull and Push	Pull	Pull
Architecture	Only master (agentless)	Master with agent	Master with agent
Configuration Language	Python, YAML	Ruby, Embedded Ruby	Ruby
Scalability	Very High	High	High
Interoperability	High	Moderate	Moderate
Cost	Low	High	High
Performance	High	Moderate to High	Moderate to High

### 6. CONCLUSION

It is important to introduce IT automation in various sectors requiring human intervention to speed up simple and complex tasks that can be done by the machine itself. In the existing education system there are many limitations related to computing, deployment and performance. This research work has proposed an DevOps based automation solution with the help of Jenkins and Ansible to overcome the existing limitations in current education system. The proposed system has a GUI based application that can serve as a system to accurately automate tasks that previously required human intervention in the education sector has thus been implemented successfully. The benefits provided by proposed system would be faster deployment of servers, automated installation of applications into the hosts and servers, providing scalability with the help of infrastructure as a code and so on.

### 7. REFERENCES

- [1] A. Katal, V. Bajoria and S. D. : 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 1-7
- [2] <https://www.jenkins.io/>
- [3] <https://coralogix.com/log-analytics-blog/the-definitive-guide-to-configuration-management-tools/>
- [4] Julio Sandobalin, Emilio Insfran and Silvia Abrahao, "End-to-End Automation in Cloud Infrastructure Provisioning", 26th International Conference on Information Systems Development (ISD 2017), At Larnaca, Cyprus, September 2017.
- [5] E. Luchian, C. Filip, A. B. Rus, I. Ivanciu and V. Dobrota, "Automation of the infrastructure and services for an openstack deployment using chef tool," 2016 15th RoEduNet Conference: Networking in Education and Research, Bucharest, 2016, pp.
- [6] Nishant Kumar Singh, S. Thakur, H. Chaurasiya and H. Nagdev, "Automated provisioning of application in IAAS cloud using Ansible configuration management," 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, 2015.
- [7] Juve, Gideon & Deelman, Ewa. (2011). Automating Application Deployment in Infrastructure Clouds. Proceedings - 2011 3rd IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2011. 658-665. 10.1109/CloudCom.2011.102.
- [8] Raja, R. & Nagasubramani, P.. (2018). Impact of modern technology in education. Journal of Applied and Advanced Research. 3. 33. 10.21839/jaar.2018.v3iS1.165.
- [9] Mr. Hatem M Wasfy et al., Dr. Tamer M. Wasfy, "The Education Sector Revolution: The Automation of Education", ASEE Annual Conference and exhibition, American Society for Engineering Education, June 2013, pp:8044.