

Devops Methods for Automation of Server Management using Ansible

Pranav T P, Charan S, Darshan M R, Girish L

Department of Computer Science & Engineering, Visveswaraya Technological University,
Channabasaveshwara Institute Of Technology, Gubbi, Tumkur,
Karnataka, India

pranavpante47@gmail.com, charan.siddu0817@gmail.com

Abstract — *Cloud has become an advanced technology in the modern field of information technology and the need to meet the growing demand of customers. Pressure on information technology organizations is constantly increasing to apply for a client in a private cloud. This change has already occurred as a large number of customers have begun direct contact with cloud vendors through this support. In addition, DevOps teams are at a very focused level now because they are responsible for automation and universal delivery and client programming. This paper focuses on the automation of customer application from the provision of environment to the delivery of the system.*

I. INTRODUCTION

The "DevOps" was intended in 2009 by Patrick Debois. The team is made up of "integrated" and "operational" development, which can provide the first place to better understand what people mean when they say "DevOps". They can be a set of practices that can include software development (Dev) and IT (Ops) functions. It aims to reduce the life cycle of system development and even provide continuous delivery of high quality software. From an academic point of view, Len Bass, Ingo Weber, and Liming Zhu - these three computer science researchers from the Commonwealth Scientific and Industrial Research Organization (CSIRO) and the Software Engineering Institute have suggested that they be described as DevOps "committed to transforming the system and incorporating it into standard production, while ensuring the highest quality"[1]. The use of DevOps will dominate technology — especially automation tools that can build infrastructure that remain more orderly and robust from life cycle structures”.

The two main objections of DevOps are:

Business plan management (ESM): Most people can get involved within the original definition of DevOps who will be the program managers. These operating technicians introduced the best ESM methods to DevOps, including configuration management, system monitoring, automation, and toolkit.

Agile Development: “DevOps is often interpreted as an Agile output - agile software development means closer customer interaction, product management.

The DevOps aims to improve communication, collaboration, and integration between software developers (Dev) and IT operations specialists (Ops) [2]. As part of DevOps, certain actions take common tools from the software development area e.g. continuous testing, quality testing, feature enhancement, and maintenance delivery to improve reliability and safety and provide cycles of rapid development and distribution. Many of the ideas involved in DevOps come from business plan management and agile software development movements. DevOps goals cover all delivery pipelines. They can include improved shipping frequency, faster market, lower decline in new releases, shorter lead time between adjustments; Immediate means recovery time (in the event of a new release or disabling the current system).

IT automation software processes and programs alternate duplicate processes and minimize manual interventions. It speeds up the delivery of IT infrastructure and applications by automating hand-held processes that required previous human touch. With IT automation, software is used to line up and duplicate instructions, procedures, or policies that save time and free IT staff with additional professional work. With the proliferation of virtualized networks and cloud services requiring fast, sophisticated delivery, automation is an important strategy to help IT teams deliver services with improved speed, consistency, and security. IT automation can be a powerful tool that will measure business, provide significant savings, and allow IT employees to work strategically instead of administrative work. The various centers of knowledge and cloud performance are often automated, resulting in faster performance. Due to the automation, IT areas can rise very quickly with a few errors and better know the needs of the business. A fully automated environment can reduce the delivery time of ready-to-produce resources from weeks to days. IT automation software can perform a variety of IT tasks and processes, from simple to complex.

The term "**Ansible**" can be a simple automated IT engine that optimizes cloud provisioning, configuration management, application deployment, intra-service orchestration, and many other IT needs. Designed for multi-tier deployment from day one, Ansible model your IT infrastructure by explaining how all your systems

integrate, rather than managing one system at a time. It does not use proxies and has no custom security infrastructure, so it is easy to use - and most importantly, it uses really simple language (YAML).

Ansible is an IT automation tool [4]. It can configure systems, run software, and configure advanced IT functions such as continuous deployment or non-rollback updates. The main objectives of Ansible are simple and easy to use. It also specializes in robust security and reliability, which includes a minimum of moving parts, the use of OpenSSH for transport, and language designed for people to circulate - even for those unfamiliar with the system, The automated language that will accurately define IT application infrastructure in Visual Playbooks. It is a flexible engine that uses Ansible Playbook. Ansible Tower is a business framework for controlling, protecting and managing your Ansible automation with UI and Restful API. Ansible will be **Simple**: Human readable automation. No special coding skills needed. Tasks executes in order.

Powerful: Application development, configuration management workflow orchestration.

Agentless: Agentless architecture, no agent to exploit or update, uses OpenSSH, more efficient and more secure.

Problem Statement : Install or update a number of standard and daily practices that are managed by systems and the IT world. And installing new software or outgoing updates with a new version is only possible with one program. Installing a point to install and update all systems under a server or all of the company's plans is time consuming and tedious. So here comes the thought and application of Ansible. Ansible is an automated IT tool [5]. You do not need to write custom code to make your systems work automatically; you write down the tasks that need to be done by writing a playbook, and then Ansible will find a way to take your systems to the state you would like them to be. Playbook files where Ansible code is written. Playbooks are written in YAML format. YAML stands for yet to another Mark-up Language. Playbooks are one of the key features of Ansible and tell Ansible what to do. A list of Ansible to-do items containing a list of tasks. Playbooks contain steps that the user wants to perform on a particular machine. Playbooks are made in order. Playbooks are the building blocks of all cases of Ansible usage.

Automation with Ansible is supposed for Linux system administrators and developers who need to automate provisioning, configuration, application deployment, and orchestration. Incorporating IT automation is significant to managing large numbers of systems and applications efficiently and consistently at scale. In this proposed system we will write Ansible playbooks to the automate tasks, and will run them to ensure servers will correctly deployed and configured.

1. To build system to automate the infrastructure using Ansible.
2. To implement an auto backup solution using Cron job.

Ansible should be very simple, reliable, and consistent in configuration management. Configurable layouts are simple definitions of infrastructure data and are both human-readable and machine-readable. All you would like to start managing the

systems can be a password or SSH (Secure Socket Shell, a network protocol) key [7]. Ansible allows you to quickly and easily use multitier applications. You will not write a custom code to make your systems work automatically; you write down the tasks that need to be done by writing a playbook, and then Ansible will find a way to take your systems to the region you would like them to be. In other words, you do not need to set up programs every time the machine is manually. When you run a playbook from your controller, Ansible uses SSH to communicate with remote hosts and execute all commands (tasks).

II. LITERATURE SURVEY

Cloud Computing: Cloud computing can be a distributed computer paradigm to provide customers with demand, computer services. Cloud users can provide reliable, accessible and up-to-date services to their customers respectively. Cloud itself contains virtual machines within the cloud providers of cloud providers. Provision is provided on top of those visible equipment. These virtual machines are provided by cloud users. A different cloud provider provides cloud services at a different level of output. Eg. Amazon EC2 [1] enables users to manage low-level data while Google App-Engine provides developers' development platforms to improve their applications. Cloud services are therefore divided into many types such as Software as a service, Platform as a service or infrastructure as a service. These services are available on the web within the entire world where the cloud operates as one access point for serving all customers. The construction of computerized computers faces the challenge of large-scale processing.

Cloud Types: There can be three types.

1. Private Cloud - this type of cloud is stored within the organization and is used for its internal purposes only. So the use model is not a big word during this situation. Many companies go through this arrangement and experts think that this is often the first step in getting a company into the cloud. Security, network bandwidth are not critical personal cloud issues.
2. Public Cloud - during this type the company hires cloud services from cloud providers on demand. Services provided to users using a computer model.
3. Hybrid Cloud - this type of cloud contains many internal or external clouds. This is a situation where a company visits a public computer public from its own private cloud.

Cloud - Type public cloud:

Public cloud or external cloud means the use of the cloud within the traditional norm. Public clouds are run by third-party companies, and applications from various clients may be integrated into cloud servers, storage systems, and networks.

The public cloud provides services to many customers. Hybrid Cloud: Hybrid clouds include both social and cloud types. This is most evident in the use of cloud storage to support Web 2.0 applications. Private Cloud: Private Cloud is built for the unique use of 1 client, which provides complete data control, security, and service quality [10]. The company owns the infrastructure and controls how applications are installed on private clouds that are usually built and managed by an IT company or cloud provider.

Cloud computing products and services.

They can be divided into 4 main categories:

1. Working as a service (AaaS)
2. Platform as a service (PaaS)
3. Infrastructure as a service (IaaS)
4. Software as a service (SaaS)

1. Application as a service (AaaS): These are the first type of cloud computing services that came into existence. Under this, the service is made available to the end user. The end user is asked to create an account with the service provider and start using the system. One of the first popular applications was a web-based email service with hotmail that started in 1996. Such services are now available online.

2. Platform as a Service (PaaS): Cloud vendors are companies that provide cloud services and products. One of the services they provide is called PaaS. Under this a computer platform like OS is provided to the customer or user on a monthly basis for rent. Some of the biggest vendors of cloud computing are Amazon, Microsoft, Google etc.

3. Infrastructure as a service (IaaS): Cloud computing vendors offer infrastructure as a service. One can access hardware services such as processors, memory, networks etc on a per-agreed basis and price.

4. Software as a service (SaaS): Software package such as CRM or CAD / CAM is usually available under the cloud computing program. Here the customer is already registered to use the software that is accessible online and for his or her business process. Work-related information can also be stored on local machines or by service providers. SaaS services can also be acquired through rental or individual use.

Cloud Computing Performance:

Cloud computing programs are usually divided into two categories: front and back. They are connected via a network, usually the web. The front is what the pc user, or client, side sees. The background conclusion is that the "cloud" section of the system. At the back there are various computers, servers and data storage systems that make up the "cloud" of computer services. The central server manages the system, monitors traffic and customer needs to ensure that everything runs smoothly. It follows set rules called protocols Servers and remote computers that do a lot of work and store data.

What is DEVOPS?

DevOps will be a set of tools that help organizations build, and use software reliably and efficiently. DevOps empowers organizations to adapt and deliver their products in addition to those with traditional development, which can provide a competitive edge Instead of chasing releases once every two weeks or more, new features can be delivered to users daily and fix bugs on delivery.

They can be essential for any business that is willing to rely on, accelerate, and respond quickly to changing markets [8].

They can be important to customers, so all stakeholders in the delivery process should need to work together. Development teams will be designing, developing, deploying and running

software very quickly. Task teams should identify and resolve the problem as soon as possible by monitoring and managing the environment and addressing issues. Integrate the standard approach across Dev and Ops by being able to monitor and analyze and perform well as quickly as possible.

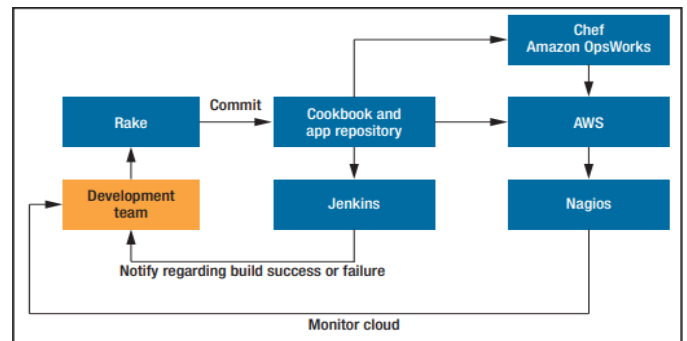


Fig 1: Devops Architecture

III. DEVOPS TOOLS

For DevOps tools, let's also review which tools are different and where they fall into the LifOps lifecycle .Describe and edit, which should focus on setting up DevOps duplication [10] and release management and release tracking. Visual tools or tool dealers in this area include Atlassian, CA Technologies, IBM, Rise, and Jama Software.

Code, build, and editing, focusing on code development and reviews, source code management, and code integration. Visual tools / vendor tools include BitBucket, Electric Cloud, GitLab, GitHub, and IBM.

- Inspection, which ensures that the quality of software and code output is maintained throughout the development process and that the highest quality is sent to production. Visual tools / tool vendors include Delphix, FlawCheck, HP, IBM, Microsoft, Parasoft, SonarSource, Skytap, and ThoughtWorks.

- Packing and recycling, referring to the activities involved when the release is ready to be shipped; also called stage or reproduction. Visual tools / tool vendors include IBM, Inedo's ProGet, Jfrog'sArtifactory, SonatypeNexus repository.

Release, deployment, and orchestration, which is a software release process and often includes change management, release authorization, automation extraction, schedule planning, provisioning, and production in production.

- Continuous management and configuration includes continuous automatic configuration, configuration management, and infrastructure such as code. Tools / vendors of visual tools include Ansible, Chef, IBM, Labs Puppet, Otter, and Salt.

- Monitors app performance reports and helps identify issues that affect user experience. Tools / tool dealers including Big Panda, IBM, New Relic, Plumb, and Wireshark [9].

IV. METHODOLOGY

DevOps has grown with Agile. Agile is a way to produce software with short iterations in a continuous schedule for the delivery of new features and bug fixes in quick cycles from two to four weeks. In contrast, DevOps brings development and working groups together to focus on eliminating silos to reduce customer

response time and break down barriers to enable continuous software delivery. As a result, they can build, test, and install software as quickly and efficiently as possible. DevOps not only engages the organization in the development process — including business lines, providers involved in software delivery, and consumers themselves — but also does it in a way that accelerates development and improves quality, according to a white paper in the new software -driven. This can lead to the creation of a innovative culture when using DevOps methods, allowing you to participate and respond strongly to market changes [8].

DevOps methods include the following:

- Continuous integration, where coding, building, integration and testing take place.
- Continuous delivery, which includes continuous integration, but is more focused on product release.
- Continuous, automated deployment of project releases very quickly.
- Works by conducting ongoing management development activities and ongoing monitoring.

A. DEVOPS PRINCIPLE

At the heart of the DevOps principles, you will find the idea of collaborative learning and the collaborative relationship between development and performance. They look to increase the speed of planned work of high export standards, while also improving the reliability, stability, durability, and safety of the production facility [11]. To establish an organization based on the principles of DevOps, you need to emphasize this comprehensive, holistic approach to all development departments and operations but also all the surrounding departments and support organization within the company. Again and again, your entire system should be used to further your organizational goals.

- Process to improve the process of reducing response logs in order to further implement the required error correction and earlier and more costly correction
- Ongoing testing that promotes self-reliance and learning from success and failure, so continuous efforts will lead to success and future success
- Students become teachers and pass on information to colleagues
- Using DevOps flexible to improve efficiency

B. DEVOPS IN REAL WORLD

Software and IT industries are all about speed and efficiency. DevOps has emerged as a paradigm for bringing products and new features quickly to market. Many technologies have recently emerged to smooth the transition between development and operation. They have the same habits of delivering fluids, thus controlling the difficulty. Fast application delivery technologies, such as looking for or selling a platform, do not work on embedded software. But we can certainly learn from such methods and map them to other domains. As mentioned earlier, the emergence of aerospace technology for faster vehicle updates indicates that targets can be transferred, while

taking into account the critical safety and operational requirements. Cloud and Web Development has been the first adoption of DevOps practices and can serve as a guide for other domains. For example, Amazon Web Services (AWS) provides several tools for continuous delivery [12]. One such tool is AWS Elastic Beanstalk, which supports continuous transmission in a simple way and therefore a shallow learning curve, but with little adjustment. AWS Ops Works offers a centralized approach, where you can write infrastructure; provides integration with Chef. You can also create an AWS Cloud Formation template, written in JSON format, to provide duplicate infrastructure and control all cloud infrastructure. You can use the AWS Code Deploy utility to install applications on multiple virtual machines (Elastic Compute Cloud Conditions) indefinitely. Alternatively, you can use the AWS Code Pipeline. This service, launched in 2015, includes construction, testing and operation. Along the way, you can use other AWS materials to support continuous delivery. AWS Code Commit is a managed resource management service that manages confidential archives. AWS Cloud Watch provides monitoring and warning infrastructure that can help the whole team work with the integrity and efficiency of the system used. Figure 2 shows the construction of DevOps using AWS tools

C. ANSIBLE

Ansible provides reliability, consistency, and robustness to your IT infrastructure. You can automatically set up information, storage, networks, fire logs using Ansible [10]. It ensures that all required packages and all other software are compatible with the server to run the application. Let's take an example; you have an app debugging version built into the C ++ visible. Now if you want to run that program on a computer, you'll need to meet the requirements for Microsoft Visual C ++ library DLLs, and you'll need a virtual C ++ installed on your computer. So, this is the part where Ansible will make sure that all these basic packages and all the software are installed on your computer so that your app can work properly in all areas, be it testing or production. It also contains all the details of your app history, so if at any time you want to go back to the previous version, or want to upgrade it, you can easily do that. Let us consider some of the following.

Agentless - Which means that there is no software or agent holding the node as another solution as cartoonist and chef.

Python - Built on python, fast and one of the strongest programming languages in today's world.

SSH - Protocol to ensure easy and secure secure network. Therefore, it is your responsibility to copy this key to the client

Push Configuration - Push the configuration required for them, clients. All you have to do is write down what is prepared (the playbook) and press them all at once in places. You can see how powerful it can be to press change on thousands of servers in minutes.

Setup - minimal need and adjustment needed for it to work.

ANSIBLE ARCHITECTURE

Let us start with Public/Private Cloud which is a Linux server. It can also serve as a repository for all IT installations and maintenance.

The design shown above also includes a number of capture devices where the portable server connects and pushes SSH playbooks. It has an accessible automation engine, which allows users to use the playbook directly on the hosts. A usable engine with a design is made of many materials. First is the list of managers. In the list of all IP addresses we host. Next, there are modules. Ansible has hundreds of built-in modules, which are pieces of code that are made while using a playbook. The playbook contains theater plays, each containing different roles, and each contains modules.

Modules are applied to your hosts while running a playbook, and these modules include interventions. As a result, when you run a playbook, every action takes place on your capture machines. You have the option to build your modules again. You just need to write a few lines of code and save it as a module, after which you can use it whenever you want. After that, there are construction playbooks.

Playbooks actually describe workflow and any activities you write in a playbook are done in the order in which they were written. It will do the same if you have written to install the package first and then start. The YAML code for playbooks is very easy to write [13].

YAML is a basic data data language very similar to English. Plugins are the next thing in architecture. Plugins are a special type of module in this case. Until a module is created for nodes, these plugins are not implemented. For login purposes, plugins are run on a large control machine. It has plug-back plugins and allows you to connect to various Ansible events for display and login. To protect expensive data collection tasks, cache plugins are used to store virtual cache.

D. HOW ANSIBLE WORKS?

Ansible works by connecting to nodes and distributing small ones organized as suitable modules. By default, Ansible will use these modules over SSH and remove them upon completion.

Ansible management node is a command and control node that monitors the entire performance of the Playbook. The management node connects via ssh, then uses the modules on the capture machines and installs the product. It removes modules after installation. But that's the way it works.

V. SYSTEM ANALYSIS

It was a challenge in the early days of networked computing to deploy and manage servers in a reliable and productive manner. Device administrators used to manually control servers, downloading software, modifying settings, and administering services on each one.

Administrators found they couldn't scale their manual system administration as quickly as the applications they were enabling as data centres expanded and hosted applications became more complex. It also hindered the developers' productivity because, while the development team was agile and constantly releasing applications, IT operations will spending more time configuring the systems. As a result, server provisioning and configuration management applications grew in popularity.

Consider the time-consuming task of managing a server fleet. We must constantly update, push updates, copy files to them,

and so on. These tasks confuse and take a long time to complete.

VI. PROPOSED SYSTEM

DevOps is a culture that promotes the development and performance of teams to work together to deliver code to production in an automated and repetitive way. The term "DevOps" is composed of the words "growth" and "performance." DevOps services to increase organizational performance and service delivery speeds. Allows businesses to provide better services to their customers and perform better in the market.

Ansible is a well-known DevOps framework for managing IT configuration. Red Hat offers Ansible, an open source automation tool and a suspension management tool. The use of YAML templates is a major advantage of Ansible as a DevOps tool. The use of automated and repetitive processes rather than ad hoc scripting or manual configuration management ensures Ansible performance.

A. ANSIBLE WORKFLOW

Ansible connects to your nodes and distributes a small programme known as Ansible modules to them. After that, Ansible ran these modules and disabled them when they were finished. There are no daemons, servers, or databases needed for the library of modules to run on any computer [10].

The Management Node is the controlling node in the above picture, and it is in charge of the playbook's entire execution. The inventory file contains a list of hosts on which the Ansible modules must be mounted. The Management Node establishes an SSH link with the host's computer and runs the small modules that install the programmed.

Once the modules have been expertly assembled, Ansible removes them.

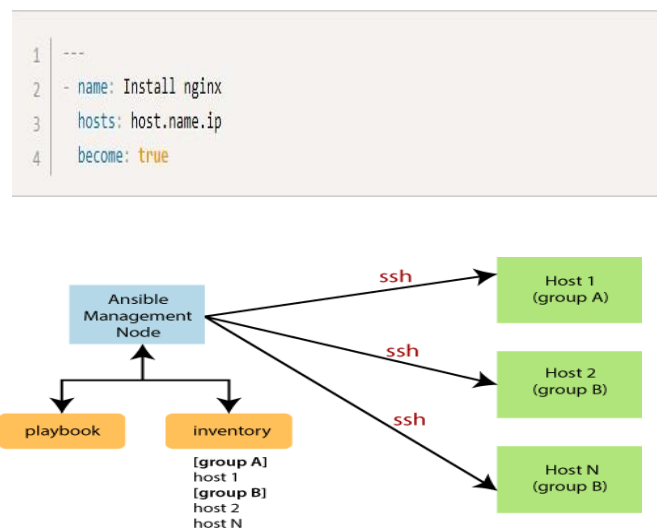


Fig: 2 Ansible Playbook

B. IMPLEMENTATION

Ansible is an automated IT engine that enables cloud provisioning, configuration management, application deployment, intra-service orchestration, and various other IT functions. Playbooks are basically pieces of commands (you play) that you send to a single target or group of indicators to be performed (host). Think of instructions for assembling equipment or furniture. The manufacturer compiles instructions so that you can assemble the parts in the correct order. When followed in sequence, the furniture looks like it was purchased.

C. HOW A PLAYBOOKWORKS ?

The Playbook we are working on will set up a web server in the RHEL / CentOS 7 server, and create an index.html file based on the model file to be included with the final Playbook.

Authors

-The author adds instructions for the modules to work, usually with additional values (arguments, locations, etc.).

-The target keeper has the opposing modules in the Playbook sequence (with inserts or other additional files).

The host status has been changed (or not) depending on the module run results, which indicate the entry of Ansible and Tower.

-In terms of furniture simulation, the Playbook is short for telling the modules to do the job. You need to understand the following in order to use your Playbooks effectively:

1. Target

Because Playbooks provide guidance and interaction modules, Ansible thinks you know how to do what you try to do and work on your own.

That's why Playbooks are like commands or directions - it tells the default sections how you want task configured [14].

2. Activities

If part of the Playbook needs to start a web server, you will need to know how to do this so that you can use the service module and start the web server by name. When Playbook installs software, you should know how the installation is performed on the target.

For example Playbook

The host will be the basic installation of RHEL / CentOS 7. The web server (NGINX) will be installed, and the index.html file

```

1 ---
2 - name: Install nginx
3   hosts: host.name.ip
4   become: true
5
6   tasks:
7     - name: Add epel release repo
8       yum:
9         name: epel-release
10        state: present
11
12    - name: Install nginx
13      yum:
14        name: nginx
15        state: present
16
17    - name: Insert Index Page
18      template:
19        src: index.html
20        dest: /usr/share/nginx/html/index.html
21
22    - name: Start NGINX
23      service:
24        name: nginx
25        state: started

```

will be installed on the default webroot. Service will be started

after the installation and file operations are completed.

Playbooks begin with the three dashes (---) of the YAML syntax, followed by:

Name: good to keep Playbooks readable

Ignorant: indicates the Ansible target that will work with it

Be the statement: a true statement is included here to confirm the installation of nginx without a problem (not always required)

```

1 tasks:
2   - name: Add epel-release repo
3     yum:
4       name: epel-release
5       state: present
6
7   - name: Install nginx
8     yum:
9       name: nginx
10      state: present

```

The tasks: statement will be placed on same indent level as the three preceding statements, after which any plays will be identified in a deeper indent (per YAML nesting). Although there are two tasks mentioned, both use the Yum module. The first Yum task is to add there repo in order to install nginx. Yum is used to

```

1 - name: Insert Index Page
2   template:
3     src: index.html
4     dest: /usr/share/nginx/html/index.html

```

install the nginx software until epel is installed.

The state: present statement instructs Ansible to first verify the state of the target before proceeding. If the repo or package is already present in either case, Ansible knows it doesn't need to do anything else for this task and continues.

Let they want to verify that nginx is installed properly, the default install page is perfect, but you may have a simple html file that you'd like to use as your proof. For the sake of convenience, I'll run the Playbook from the same directory as the template index file. With no configured pages, the destination is simply nginx's default.

The Playbook will ensure that the nginx service has been started as the final phase (and if not, start it). The entire Playbook is about the same length as the introduction

VII. CONCLUSION

Infrastructure design is a software life cycle step that defines and tailors software infrastructure requirements, including the number and form of material managers or required hardware. Infrastructure construction often requires a large amount of installation and maintenance documentation required to do things like: (i) strengthening and connecting the equipment needed (either physical or physical) for the software to work, (ii) installing and repairing the required software and purchases within the virtual machine, (iii) to strengthen and run the support resources needed for the software to be used. We have developed a new layer of the Ansible program, representing the framework of orchestration and configuration management, in line with current Infrastructure-like codes. Ansible's new platform enables Brno University of Technology in the Czech Republic to manage university labs from a local and community network.

In addition, a new web interface has been developed to make it easier to customize daily activities - a customized app written in Java and using the Spring framework. Machine management from the university's internal network and social network was fully tested with a web-interface, and all software upgrades were successfully considered, as playbooks are intended to be readable to the individual. Once the stated goals have been achieved, equipment management from the university's internal network and social network is fully tested via a web interface, and as a release, it can be ensured that all required software changes are successfully accepted. is a useful way to secure remote control and configuration of selected network objects.

REFERENCES

- [1] Elastic Compute Cloud(EC2), Amazon Web Services, Amazon. com, <http://aws.amazon.com/ec2>.
- [2] FutureGrid, <https://portal.futuresystems.org>. W. Gentsch, Sun Grid Engine: towards creating a compute power grid, 1st IEEE/ACM Int. pp. 35-36, May2001.
- [3] G. Juve, E. Deelman, K. Vahi, and G. Mehta, Scientific Workflow Applications on Amazon EC2, 5th IEEE Int. Conf. on e-Science Workshop (e-Science 2009), pp. 59-66, Dec 2009.
- [4] Girish L, Rao SKN (2016) Mathematical tools and methods for analysis of SDN: a comprehensive survey. In 2nd international conference on contemporary computing and informatics (IC3I), Noida, pp 718-724. <https://doi.org/10.1109/IC3I.2016.7918055>
- [5] C. David, Introduction to DevOps on AWS, AWS, pp. 2, Dec. 2014 <https://d0.awsstatic.com/whitepapers/AWSDevOps.pdf>
- [6] Sentinel Cloud Services, Run-time Guide, Safenet, 2014
- [7] Sentinel Cloud Service, Sentinel Cloud V3.5 Quick Start Guide, Sentinel, Jun 2014
- [8] Automated Provisioning and Deployment, Vmware WhitePapers,
- [9] Rashmi T V. "Predicting the System Failures Using Machine Learning Algorithms". International Journal of Advanced Scientific Innovation, vol. 1, no. 1, Dec. 2020, doi:10.5281/zenodo.4641686.
- [10] Ansible, <http://docs.ansible.com>.
- [11] VLDB Endowment, vol.2, Iss.1, pp. 253-264, Aug. 2009.
- [12] P. Shivam, A. Demberel, P. Gunda, D. Irwin, L. Grit, Automated and On-Demand Provisioning of Virtual Machines for Database Applications, In ACM Proc. Int. Conf. Of Management of Data (SIGMOD 2007) pp. 1079-1081, 2007
- [13] Girish, L., Rao, S.K.N. Anomaly detection in cloud environment using artificial intelligence techniques. Computing (2021). <https://doi.org/10.1007/s00607-021-00941-x>
- [14] L, G. (2019). Anomaly Detection in NFV Using Tree-Based Unsupervised Learning Method. International Journal of Science, Technology, Engineering and Management - A VTU Publication, 1(2), 27-31. Retrieved from <http://ijesm.vtu.ac.in/index.php/IJESM/article/view/232>