

**Network Process Automation: How Ansible Can Facilitate Efficient Processes**

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science  
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree  
Bachelor of Science, School of Engineering

**Samarth Saxena**

Fall, 2022

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Mark Sherriff, Department of Computer Science

**Commented [SF1]:** Insert your advisors name here.

If you have more than one, please add another below this and continue adding advisors until you are complete.

# **Network Process Automation: How Ansible Can Facilitate Efficient Processes**

CS4991 Capstone Report, 2022

Samarth Saxena  
Computer Science  
The University of Virginia  
School of Engineering and Applied Science  
Charlottesville, Virginia USA  
[ss9nfb@virginia.edu](mailto:ss9nfb@virginia.edu)

## Abstract

The Federal National Mortgage Association, more commonly known as Fannie Mae, discovered that the system it used to map and test networks was complex, time-consuming, and inefficient. To solve this problem, I decided to use an automation platform named Ansible, in conjunction with Cisco Modeling Labs (Cisco CML) and Python. The results of the project were successful; a several-hour long process was shortened to 15 minutes. My internship ended before I could do the work myself but the next steps that are needed would be to thoroughly test the project to ensure that there are no bugs.

## 1. Introduction

“Automation applied to an inefficient operation will magnify the inefficiency.” – Bill Gates. This quote directly applied to the network system that was in place at Fannie Mae when I first started my internship. The overall foundation of the system was inefficient, with little reuse of code and use of functions. There was also no way to quickly change input parameters on tests without manually changing the code. These issues caused large amounts of downtime—ranging anywhere from 30 minutes to 2 hours—between running multiple tests. My goal was to minimize the downtime.

## 2. Related Works

I did not do much research; I talked to people in my organization about what tools they would use in my situation and took notes. Specifically, I had in-depth conversations with my manager and mentor and found that Ansible software was commonly utilized. After I found that it was, I used a LinkedIn learning class on Ansible to familiarize myself with the tool and was able to learn how to create a playbook in Ansible, how to automate procedures, and how to connect to a specific IP address in Ansible [1]. Ansible uses Python, so I researched Python by reading some of the documentation [2].

Figure 1: Example Ansible Playbook

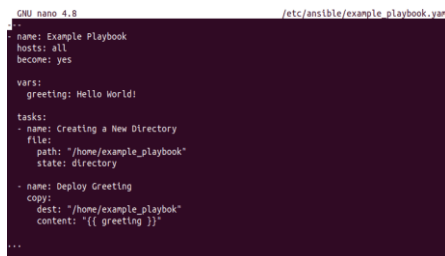
## 3. Project Design

Ansible utilizes playbooks, which allow users to manage servers remotely. Playbooks are YAML files that contain a list of tasks to be executed on a remote server [1]. The first step of my project was to create playbooks that connected to the server on which the network tests were run. To do this, I made the IP address of the server the host of the playbook.

The next step was to create a task that could create a copy of the test. This is necessary so the playbook could modify the test in the manner that the user wants. I did this by creating a playbook that would download the test as a YAML file and save that file as a new playbook.

The final step of my project was to create tasks that would modify the playbook as the user wanted. I completed this by making tasks that would receive the input parameters the user wanted and create a new copy of the YAML file that was downloaded, except with the input parameters changed. This would give the user the illusion that the original test was modified and run.

## 4. Results

A screenshot of a terminal window showing an Ansible playbook file. The terminal title is 'GNU nano 4.8 /etc/ansible/example\_playbook.yaml'. The content of the file is as follows:

```
- name: Example Playbook
  hosts: all
  become: yes

  vars:
    greeting: Hello World!

  tasks:
    - name: Creating a New Directory
      file:
        path: '/home/example_playbook'
        state: directory
    - name: Deploy Greeting
      copy:
        dest: '/home/example_playbok'
        content: '{{ greeting }}'
...

```

The results of my project are apparent when employees tested their network systems. The downtime between running multiple tests used to take anywhere from 30 minutes to two hours on average. Now, the downtime is about two minutes, with a maximum of five minutes. Employees now can modify an Ansible script to change input parameters, rather than having to manually change input parameters in the code.

## **5. Conclusion**

My project's goal was to minimize the downtime that occurred when consecutive network system test cases were ran. I created playbooks that automatically ran tests, opened websites, ran functions, and changed input parameters, based on user input. I was able to successfully achieve my goal: downtime decreased from two hours to two minutes.

## **6. Future Work**

Unfortunately, my internship ended before I was able to complete the full lifecycle of the program I had created. Future work needed on my project includes thoroughly testing my program to ensure that there are no glitches or bugs present in the code. In addition, there are certain features that I would like to add to the playbooks, such as the ability to automatically run multiple tests concurrently; for example, running a whole suite of tests at once.

## **7. UVA Evaluation**

UVA's CS program prepared me well for a career in computer science. The program instilled great teamwork abilities, communication skills, and time management skills. As far as technical skills are concerned, I used almost nothing that I learned at UVA on the job. However, I expected this to some degree, as every company has its own environment that must be learned. It would be impossible for the CS program at UVA to teach every single tool that could be used professionally.

## **8. Acknowledgments**

I would like to thank my manager, Daniel Negussie, my mentor, Oleksii Evdokimov, and a good friend/mentor during the internship, Griffin Cosgrove. Without their help and advice, I would not have been able to accomplish my goal of lowering the downtime between tests.

## **References**

[1] Anthony Sequeira. 2020. Starting your Ansible journey - Ansible Video Tutorial | LinkedIn . LinkedIn Learning. Retrieved November 30, 2022 from <https://www.linkedin.com/learning/learning-ansible-2020/starting-your-ansible-journey>

[2] Python Software Foundation. 2022. 3.11.0 Documentation. Python. Retrieved November 30, 2022 from <https://docs.python.org/>