Using Scripts to Improve the Efficiency of Developers and the Usability of Previous Developments

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

Qinyuan Jiang

Spring, 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

Briana Morrison, Department of Computer Science

Rosanne Vrugtman, Department of Computer Science

Using Scripts to Improve the Efficiency of Developers and the Usability of Previous Developments

CS 4991 Capstone Report, 2022

Qinyuan Jiang
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
qj8eu@virginia.edu

Abstract

Global financial services company often has developers who have to access another developer's code to test or complete a task at hand. However, it can be difficult to know what has already been developed and where the code is prior to starting a new task. To combat this issue, I developed scripts along with other developers on my team. To make different developers' codes more accessible, I uploaded code files onto GitHub to be tested on Ansible Tower with automated tests. Then I wrote scripts based on different issues that developers have come across and code that has been developed by others.

As a result of this work, other development teams within the company can easily perform tasks reliant on these new updates. In addition, developers will be able to implement code without having to worry about whether it will be used in the future, because these pieces of code will be uploaded onto GitHub. In the future, more work can be done to make developed code accessible throughout the company using a centralized platform for all departments and updating the platform regularly.

1 Introduction

A common coding concern among developers is how big an impact their code will have or whether their code will be used at all. If not used, will the work they have done been for nothing? So why is their code not used? The code must have been created to solve a problem at hand.

This was a common concern for the developers in the team that I interned with. Sometimes developers' scripts will just sit there for months without any action to apply them to the applications or servers they were written for. Considering that the company I interned at is not as technically focused as other companies, this may be a more prominent issue at companies that have developed much more complex code that will take longer to implement.

If this concern is addressed, developers will have more confidence to create efficient code without feeling that their work will not have an impact on the company and the technology field. Instead, they will have a sense of accomplishment after spending hours and overcoming challenges in the development process. My approach to addressing this problem was

implementing scripts, along with another developer, to utilize others' code to solve problems with applications and servers within the department where I interned.

2 Related Works

Since the team I was in widely used Ansible for the support we provided, I had to familiarize myself with Ansible, which Red Hat (2022) defines as an "IT automation engine that automates provisioning, configuration management, application deployment, orchestration, and many other IT processes" [3]. Ansible playbooks are written in YAML, a data serialization language. I also used a lot of documentation articles to help me understand the language's syntax. Another Red Hat article (2021) provided a basic overview of what YAML is used for and helped me to understand its syntax [6]. On top of doing research online, I also gained help from other developers in understanding Ansible and YAML.

For the tasks that I helped to complete within my team, I used GitHub, Ansible, and YAML to resolve issues that other developers have come across. Some of these tasks include restarting servers, scheduling tasks, and installing packages onto servers. These tasks all had code that has been developed by other developers.

3 Process Design

Since I was never given a project to work on for the duration of the summer, I worked on smaller tasks instead. The process that I used to complete these tasks are the same. I would develop the YAML code on Visual Studio Code, push the code onto GitHub, then run the repository on Ansible Dev Tower or somewhere else to test and debug my code.

3.1 CRON Job Scheduler

My first task was to upload an already implemented and tested Shell script, which found and updated CRON entries for self-healing, onto GitHub and then run this code in YAML for different servers.

The purpose of CRON is to "schedule commands for automatic execution" at a given time [5]. To do this, the YAML code would check if the Shell script is in the GitHub repository. Then, using YAML commands, I would copy that script onto the given server and remove it after everything within the shell script had been executed. This code was tested on Ansible Dev Tower on ten different servers.

3.2 Puppet Convergence

When it comes to software, it is important to keep applications up to date. The same can be said about Puppet, a tool used to "manage and automate the configuration of servers" [2]. The goal of this task was to update Puppet packages and modules on targeted servers.

To accomplish this goal, I took a similar process using YAML. Before starting this task, I was already given the Python script, developed and tested by another, for applying the releases and migrate nodes from different versions of Puppet environments. In this case, the nodes are blocks of code [4]. When writing the YAML file, I made sure that the provided Python script exists within GitHub and can be copied to the host server. After the script had been executed, it was removed from the server. This YAML code was, then, tested on ten Unix servers on Ansible Dev Tower.

3.3 Apache Restart

For this task, I restarted a specific service, httpd, on Unix servers. Httpd is the "Apache HyperText Transfer Protocol

(HTTP) server program" [1]. Since the code to restart this service had been provided in another file, I made sure that my YAML code validated that it existed first. Then, I executed the file, killed any future processes and waited for the already running processes to finish and start Apache again. This file was only executed on two test servers on Ansible Dev Tower.

3.4 Self-Healing Package

I was tasked with installing a self-healing package on Windows servers. This package was provided in the form of a zip file on GitHub. To access this zip file in YAML, I created a directory to copy the zip file onto the new directory. Then, I unzipped it in the same directory and ran the self-healing install script within the package. After the script had outputted something indicating that it had finished running, the zip file was removed to clear space. To test this YAML file, something different was done in addition to running it on Ansible Dev Tower. I tested this YAML file on an actual Windows server by another developer by directly logging into the server and executing the file.

4 Results

In these smaller tasks I was given, I was able to connect other developers' code onto a centralized platform, so that more developers could access them to deploy code onto the servers of their choice. This enabled more developers' code to be more easily accessed in the future. It changed the process of running someone else's code, taking it from contacting individuals to see if there was code already developed and searching through an unfamiliar GitHub repository to going on a centralized platform and entering the server on which to deploy the code. This significantly shortened by 30% the time it took to complete tasks.

5 Conclusion

I developed different pieces of code that can access other developer's code on a desired server and made the process of development easier for other developers. If widely used within the company, this process will be helpful not only for the department I was a part of, but all technical departments and developers.

6 Future Work

In the future, in order to make more developers' code available on a centralized platform, code can be developed to ask the user for the file location and server name. The code will take these input values and find the file location, making sure it exists in that location, then execute the commands within that file on the server specified. This makes it easier so that new files do not have to be created for each piece of new code developed.

References

[1] The Apache Software Foundation. 2022. httpd - Apache Hypertext Transfer Protocol Server. Retrieved October 18, 2022 from https://httpd.apache.org/docs/2.4/programs/httpd.html

[2] Puppet Inc. 2022. Introduction to Puppet. Retrieved October 18, 2022 from https://puppet.com/docs/puppet/6/puppet_overview.html

[3] Red Hat. 2022. Learning Ansible basics. Retrieved October 18, 2022 from https://www.redhat.com/en/topics/automation/learning-ansible-tutorial

[4] Puppet Inc. 2022. Node definitions. Retrieved October 18, 2022 from https://puppet.com/docs/puppet/6/lang_node_definitions.html
[5] Pair Knowledge Base. 2022. What is Cron and How do I Use It? Retrieved October 18, 2022 from https://www.pair.com/support/kb/configuring-cron/
[6] Red Hat. 2021. What is YAML? Retrieved October 18, 2022 from https://www.redhat.com/en/topics/automation/what-is-yaml