Deploying Big Data and Development Environments Using Ansible

Hagen Hodgkins
Elizabeth City State University
1704 Weeksville Rd
Elizabeth City, NC 27909
hagen.hodgkins@gmail.com

Badi' Abdul-Wahid
School of Computing and Informatics
Indiana University
Bloomington, IN 47408
abdulwahidc@gmail.com

Gregor von Laszewski
School of Computing and Informatics
Indiana University
Bloomington, IN 47408
laszewski@gmail.com

ABSTRACT

When developers need to set up a new machine with the intent to use it as a development environment they must go through the hassle to create a list of what they need. They must then proceed to individually download each of the programs and dependencies that they would require for their machine to be viable for the development of projects. The creation of an automated system for downloading a preexisting list of programs and dependencies would allow a developer to skip this step entirely and relatively quickly delve into their development tasks. With this in mind we applied DevOps concepts and worked with ansible to create a role that did just this. We then applied the benefits of ansible role automation to a big data project which had several roles built into it that otherwise should be absent from the project's repository. As such we created individual repositories to host the roles and created an ansible document within the project that would call these roles should they be needed.

Categories and Subject Descriptors

D.3.3 [**Programming Languages**]: YAML

General Terms

Algorithms, Management, Documentation, Performance, Design, Reliability, Experimentation, Standardization, Verification.

Keywords

Ansible, Big Data, Cloudmesh, FutureSystems, Linux, Python,

1. INTRODUCTION

DevOps is a term for a group of concepts that emerged from the two related trends in technology. The first is “agile system administration” or “agile operations” The second is an expanded understanding of the value of collaboration between the development and the operations staff throughout the entirety of the development lifecycle when creating and operating a service. We used these concepts in our projects. DevOps has received a good deal of attention in recent years as is shown by the release of framework platforms that assist those working in DevOps. For our projects we used the Ansible framework platform to assist with automation efforts.

The primary project was the creation of a way to automate the setup of a python development environment for a Linux machine running Ubuntu Xenial 16.04. For the purpose of creating an automated deployment of programs and dependencies we chose to use Ansible. Ansible allowed us to create a role that could be downloaded and deployed in very few steps to install all the necessary programs for a developer to begin working with python. Ansible playbooks and roles are coded in the YAML language which is relatively easy to learn with is almost pseudo code structure. The role was successfully created and several of its playbook were used in the setup of a development environment on another groups machine.

The secondary project our team tackled was the cleanup and application of DevOps concepts to a big data project dealing with the comparison of populations and unemployment rates in counties throughout the United States. When initially observing this project we noticed that the project had its roles built into the projects repositories rather than hosted in their own repositories and called into the project when downloaded and deployed. This was an issue that we sought to quickly resolve by moving these roles to their own repositories and creating an Ansible playbook to call the roles. The readme files within the project also saw an overhaul as they had either irrelevant information or incorrect information. However, this project needs additional work as certain aspects of the documentation are still incomplete and the roles do not function correctly due to either issues with permissions or syntax errors.

2. DEVOPS FOR BIG DATA

DevOps is a term for a group of concepts that, while not entirely new, have been combined into a movement that is rapidly spreading throughout the technical community. DevOps is a relatively new term emerging from the collision of two related trends. The first was called “agile system administration” or “agile operations”; it originated from applying newer, more agile, approaches to operations work. The second is an expanded understanding of the value of collaboration between the development and the operations staff throughout the entirety of the development lifecycle when creating and operating a service. [5]

3. DevOps Frameworks

3.1.1 Ansible

Ansible is an automation engine that allows a user to automate a number of different tasks such as cloud provisioning, configuration management, application deployment, intra-service orchestration, along with many other IT related tasks. Ansible is designed to carry out multi tiered deployments. It uses the YAML language to allow users to create what are called playbook to organize and create their automation system. [4]

We used ansible due to the relative ease with which users can create and deploy roles. In addition Ansible provides hosting for users to upload and share roles via Ansible Galaxy. This allows a team to separate roles housing dependencies from the main project repository but still apply them to the project when needed. We utilized this feature extensively throughout the projects that the team undertook.

3.1.2 Puppet

Puppet Enterprise manages infrastructure as code, providing the foundation for DevOps practices such as versioning, automated testing and continuous delivery. Developers deploy changes with confidence and recover more quickly from failures, freeing your team to be more agile and responsive to the needs of the project. [6]

3.1.3 Chef

Chef is a platform that has several different ways to help developers. Chef allows developers to write dynamic policies that automatically create and configure infrastructure when you need it. Developers can search the entire infrastructure at any time–and use real-time data in your policies. It also allow for developers to automatically deliver the latest tested and approved policies to your infrastructure. Developers can use chef to manage complexity with a scalable automation platform. [2]

3.1.4 Saltstack

SaltStack software helps DevOps groups by orchestrating the movement of code into production and by keeping complex infrastructures fine-tuned for optimal service and application delivery. DevOps may not be a tool, but it does require a tool-centric approach to realize value in continuous delivery. SaltStack orchestrates the DevOps process and helps developers to deploy and configure dynamic applications, and the general-purpose infrastructure they run on. [8]

3.1.5 Docker

"Docker is an open platform for developers and system administrators to build, ship, and run distributed applications." [9] The goal of Docker is to act like an ordinary shipping container if a shipping container was intended for software. The developer packages a container with the required software and its dependencies. The DevOps and infrastructure team manages and deploys the container. [9] Docker containers wrap a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries – anything that can be installed on a server. [10]

4. INFRASTRUCTURE FOR BIG DATA

4.1.1 Clouds

Cloud computing is storing and accessing data and programs over the Internet rather than from a computer's hard drive. The cloud is just a metaphor for utilizing the Internet. Everything tangible that one might need is close to the user physically, this means accessing the data is relatively quick and simple for computers on that network. For it to be considered cloud computing, users need to be able to access data or programs over the Internet, or at the very least, have that data synced with additional information located over the internet. In a big business, users may know all there is to know about what's on the other side of the connection; as an individual user, you may never have any idea what kind of massive data processing is happening on the other end. The end result is largely identical: with an internet connection, cloud computing can be done anywhere, anytime, with ease. [3]

4.1.2 Containers

Containers provide a solution to the problem of how to get software to run reliably when shifted from one computing environment to another. This could possibly be from a developer or researcher's laptop to a test environment, or even from a staging environment into production as well as from a physical machine in a data center to a virtual machine in a private or public cloud. [7] A container is an isolated environment such that one or more processes can be run. Containers focus on process isolation and containment instead of emulating a complete physical machine. [9]

4.1.3 High Performance Computers

Big data is an environment that is almost entirely reliant on the ability to store and process information in almost incomprehensibly vast quantities. As such the ability of a computer to handle the strain of these actions and in a rapid manner is critical for the machine to be viable for big data. A machine that cannot handle the strain or cannot go through the information quickly will lead to issues is big data science where researchers are sitting around waiting for a process to finish because it cannot handle the flow quickly or otherwise looking for replacements because it cannot handle the flow at all.

5. DEVELOPMENT ENVIRONMENT FOR UBUNTU XENIAL

5.1 Development methodology

This project focuses on giving a developer on a new machine the environment he needs to begin development using python and working with Cloudmesh. For this purpose the team decided to use an Ansible playbook. When deployed the playbook would proceed to install python, its dependencies, a variety of python editors, as well as Cloudmesh client. The majority of the tasks in the role were written into a file called python.yml which was responsible for installing all but the Pycharm application, which in turn was written into a separate .yml file due to more specific instructions required for its automatic installation. Both of these are called from main.yml which the user launches to deploy the playbook on their machine. The main.yml file uses include functionality to call and launch the other .yml files automatically, thus deploying the playbook and installing the programs.

A readme file was created in the primary directory of the role which outlines what the role does as well as the instructions on how to deploy the role onto a new machine. The role was then tested piecemeal as it was created to ensure that the components worked properly before rolling them together and testing the project as a whole.

5.2 User instructions

The development environment setup for Ubuntu Xenial uses an Ansible role hosted on Ansible Galaxy. This role is responsible for installing a host of different programs which are either required or otherwise beneficial for a developer working with Cloudmesh and python. This role is intended to be run on machines using Ubuntu Xenial version 16.04. Prior to deploying the role on a new machine the user will be required to install Ansible on their machine in order for the role to function correctly.

```
$ sudo apt-get install ansible
```

Once the user has installed ansible they can download the role from ansible galaxy. The user will either want to create a directory or otherwise have one prepared for the role to be installed to avoid the issue of installing the role to an incorrect directory by not being able to specify one as the install path.

```
$ sudo ansible-galaxy install cloudmesh.ansible-cloudmesh-ubuntu-xenial --roles-path=~/directory/of/your/choosing/
```

Now that the user has installed the role on the machine they wish to set up the development environment on they will need to deploy the role in order for the automated process the role provides to begin. This can be done by locating and running the main.yml file within the install directory for the role.

```
$ sudo ansible-playbook ~/directoryofyourchoosing/ansible-cloudmesh-ubuntu-xenial/tasks/main.yml
```

5.3 Results
When deployed the Ansible role should install python, a number of dependencies, as well as several python editors and Cloudmesh client. The Role itself was tested on the machine it was created on and seemed to run correctly. The python.yml was used to set up another group members development environment on their machine.

5.4 Recommendations for future works
However, there have been numerous issues encountered in regards to permissions and syntax as such it is recommended that future researchers working with this role further test and refine it. Future researchers should also consider altering the role such that it will function on other operating systems such as windows and CentOS 7

6. BIG DATA ENVIRONMENT
6.1 Population Big Data Environment
This project attempts to uncover some of the key factors behind why population numbers change. We highlight population numbers by county and use statistical data to discover trends. Data sets were made publicly available by the U.S. Census Bureau and the United States Department of Labor: Bureau of Labor Statistics. The first data set is the population change for counties in the United States and Puerto Rico: 2000 to 2010. The other two data sets are unemployment rates in counties in the United States and Puerto Rico, one for 2000 and the other for 2010. We Plan to derive meaningful insight between population changes in counties and unemployment rate, specifically how unemployment rates appear to affect population changes. The data sets are loaded into MongoDB. The analysis is conducted by a Python script, titled "analysis.py" which pulls data directly from MongoDB, does a short analysis, and exports the data in CSV format. Visualization is done on Tableau. [1]

6.2 Methodology
When we looked at this project it was immediately clear we needed to future proof the project to an extent as well as clean it up. Initially the roles were built into the project repository. These are roles that are required for other projects and as such there would have been redundancy in the roles when attempting to install this project onto a machine which hosted other projects requiring the same roles. To resolve this issue we looked to Ansible. The use of Ansible automation is important to this big data environment because it allows for the installation of the prerequisites for a project to function properly automatically and in bulk. It allowed the team to create and upload a single version of a role that we can use across a variety of projects and update as required without the need to check every project repository for duplicates of a role. This is because the role is not located within any of the projects. The roles are stored within their own repositories then called from Ansible galaxy when needed via terminal or command line onto the machine it is needed on.

This removed the need to install dependencies one by one. In the case of the population project the prerequisites are Hadoop, a framework for data storage and cluster management, and mongodb, an open source database system. This project had the ansible roles for installing the dependencies built into the projects repository. This is an issue because if done in other projects, over time it can lead to the oversaturation of redundant or duplicated roles. While initially this may not present issues other than the unnecessary consumption of storage space on the users machine as software gets updated and old syntax's become obsolete or otherwise unusable in the new version all these redundant roles will need to be independently to ensure all the projects they are used in don't become nonfunctional.

The team also took time to investigate the project as a whole to remove documentation that was not required or was otherwise unnecessary. We also ensured what documentation remained was helpful and relevant to any user who is working with the project in the future. This included rewriting the instructions for installation and setup by the user as well as additional documentation pertaining to the what the project is intended to accomplish and the process through which it is achieved.

6.3 Results
Two Ansible roles were developed and hosted on Ansible Galaxy that are intended to install and deploy both Hadoop and mongodb, the prerequisites of the population project. A majority of the work of standardizing and updating the project focused on the population1 directory of the project. The readme files saw numerous changes directed at making it more comprehensive for a new user as well as removing content that was unnecessary, this primarily pertains to research and results that weren't important to the use and description of the project.

Several important issues were encountered in the process of testing the new roles and the project itself. The platform that the project was originally recommended to run on, FutureSystems, had numerous permissions and connectability issues that we could not resolve and proved to be a major roadblock to further development of the project.

6.4 Recommendations for future works
It is recommended that future researchers have the proper permissions to run the project, such as sudo rights. Otherwise future researchers and developers should look to either bypass this issue or find another platform on which to run the project.

Future researchers and developers should continue to refine the project with a focus on making it understandable to someone who is seeing it for the first time. This includes and overhaul of the install and running instructions and the organization of files within the directory. There are many files which could be considered either redundant or otherwise unnecessary which could be either removed or edited to make them relevant to the project.

7. ACKNOWLEDGMENTS
Our thanks to ACM SIGCHI for allowing us to modify templates they had developed. Thanks to the Center of Excellence in Remote Sensing Education and Research and Indiana University for this Research Experience for Undergraduates program. Thanks to Dr. Linda Hayden, and Dr. Lamara Warren for help in securing this research opportunity. Our thanks to Gregor von Laszewski, and Badi' Abdul-Wahid for their assistance in completing this research experience.

8. REFERENCES
[1] Eden Barnett, Priyanka Jadhav, and Jeff Sustarsic. 2016. cloudmesh/ansible-cloudmesh-population. (May 2016). Retrieved June 28, 2016 from https://github.com/cloudmesh/ansible-cloudmesh-population/tree/master/population/class/population1

[2] Chef – Automate Your Infrastructure | Chef. Retrieved July 14, 2016 from https://www.chef.io/chef/

[3] Eric Griffith. 2016. What Is Cloud Computing? (May 2016). Retrieved July 18, 2016 from http://www.pcmag.com/article2/0,2817,2372163,00.asp

[4] How Ansible Works | Ansible.com. Retrieved June 5, 2016 from https://www.ansible.com/how-ansible-works

[5] Ernest Mueller. 2010. What Is DevOps? (August 2010). Retrieved July 10, 2016 from https://theagileadmin.com/what-is-devops/

[6] Puppet - The shortest path to better software. Retrieved July 13, 2016 from https://puppet.com/

[7] Paul Rubens. 2015. What are containers and why do you need them? (May 2015). Retrieved July 11, 2016 from http://www.cio.com/article/2924995/enterprise-software/what-are-containers-and-why-do-you-need-them.html

[8] SaltStack automation for CloudOps, ITOps & DevOps at scale. Retrieved July 16, 2016 from https://saltstack.com/

[9] Anand Mani Sankar. 2015. Containers (Docker): A disruptive force in cloud computing. (March 2015). Retrieved July 20, 2016 from http://anandmanisankar.com/posts/container-docker-paas-microservices/

[10] 2015. What is Docker? (2015). Retrieved June 17, 2016 from https://www.docker.com/what-docker