

**FRAMEWORK FOR DEVELOPING AUTOMATED INFRASTRUCTURE AND
SOFTWARE DEPLOYMENT FOR DOMAIN SPECIFIC RESEARCH**

PREVIEW

APPROVED BY SUPERVISING COMMITTEE:

Jeff Prevost, Ph.D., Co-Chair

Ram Krishnan, Ph.D., Co-Chair

Wonjun Lee, Ph.D.

Accepted:

Dean, Graduate School

Copyright 2016 James O. Benson
All Rights Reserved.

PREVIEW

DEDICATION

This thesis is dedicated to my wife, Susi, for her unending support and patience. To Henry and his baby sister, may you always keep your youthfulness, curiosity, and love for life and learning. As Henry David Thoreau said, go confidently in the direction of your dreams and live the life you have always imagined.

PREVIEW

PREVIEW

**FRAMEWORK FOR DEVELOPING AUTOMATED INFRASTRUCTURE AND
SOFTWARE DEPLOYMENT FOR DOMAIN SPECIFIC RESEARCH**

by

JAMES BENSON, M.S.

THESIS

Presented to the Graduate Faculty of
The University of Texas at San Antonio
In Partial Fulfillment
Of the Requirements
For the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT SAN ANTONIO
College of Engineering
Department of Electrical and Computer Engineering
August 2016

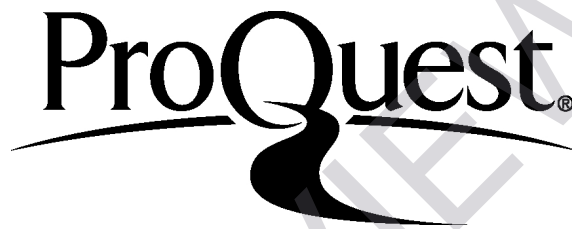
ProQuest Number: 10151218

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10151218

Published by ProQuest LLC (2016). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis committee Dr. Jeff Prevost, Dr. Ram Krishnan, and Dr. Wonjun Lee for all of their support, guidance, constructive criticism, patience, and encouragement throughout the process of writing this thesis. Further, I would like to particularly express my thanks towards Dr. Jeff Prevost, for being an incredible supervisor and trusting in me while I reach my potential in this new field. I would like to thank The Open Cloud Institute and Dr. Paul Rad for the opportunity to delve further into OpenStack with unique training from Intel and Rackspace and the OpenStack certification that pursued.

Lastly, I would like to thank my wife for all of the love, support and patience she has shown me throughout the years. You've always stood by my side no matter what. I look forward to all of our future journeys together.

August 2016

FRAMEWORK FOR DEVELOPING AUTOMATED INFRASTRUCTURE AND SOFTWARE DEPLOYMENT FOR DOMAIN SPECIFIC RESEARCH

James Benson, M.S.
The University of Texas at San Antonio, 2016

Supervising Professor: Jeff Prevost, Ph.D., Co-Chair

For the past 10 years the Cloud has been growing steadily. While the Cloud has been utilized heavily in industry, its application for research in academic settings has been limited. This under-utilization is primarily due to the lack of time of scholars, lack of funding, and lack of technical expertise to set up a cloud and research platform to do analytics.

The primary aim of this thesis research was to develop a framework that could aid researchers in academic settings in utilizing the benefits of cloud based research environments. The Cloud RAINS framework consists of an easy to use web-based interface which allows researchers to choose between pre-specified research domains and infrastructure properties. The framework then automatically builds a cloud environment based on these specifications using OpenStack, Ansible, and Python scripting. The framework makes it possible to deploy a set of tools used in a variety of research settings. These tools can be deployed without the researchers needing the knowledge of how to set up networking, start virtual instances, assign floating IP's, greatly improving access to cloud based computing. The only prerequisite for the framework is a basic understanding of what tools researchers would like to utilize in their new cloud environment.

TABLE OF CONTENTS

Acknowledgements	iv
Abstract	v
List of Tables	viii
List of Figures	ix
Chapter 1: Introduction	1
1.1 Motivation	4
1.2 Contribution of this thesis	5
Chapter 2: Background	6
2.1 A Brief Overview of Cloud Computing and Virtualization	6
2.2 An Overview of Scripting Tools	9
2.2.1 Ansible	11
2.2.2 Chef	12
2.2.3 SaltStack	13
2.2.4 Infrastructure Deployment	13
2.2.5 OpenStack Heat & AWS CloudFormation	14
2.3 Current Technologies	15
2.4 Conclusion	16
Chapter 3: Creating a Deployment Methodology	17
3.1 Comparing User Interfaces	17
3.2 The Old (Manual) Method of Getting Started	21
3.3 How to Develop A New Architecture Framework - Cloud RAINS	22

3.4	Summary	26
Chapter 4: Current and Future Security Features of the Cloud RAINS Framework . .		27
4.1	SQL Database Security	27
4.2	Password Security	28
4.3	Direct File References	30
4.4	SSL Encryption Between User and Framework	30
4.5	SSL Encryption Between Framework and OpenStack	31
4.6	Conclusion	31
Chapter 5: Conclusion		32
5.1	Research Contribution	32
5.2	Limitations	33
5.3	Future Research Directions	34
5.4	Conclusion	35
Appendix A:		36
Appendix B:		39
Appendix C:		49
Bibliography		53
Vita		

LIST OF TABLES

Table 2.1	Comparison of Cloud Providers Specifications.	9
Table 2.2	Cloud Service Type Comparison	10

PREVIEW

LIST OF FIGURES

Figure 2.1	Gartner: Magic Quadrant for Cloud Infrastructure as a Service	7
Figure 2.2	Service Model Types in order of the dynamic nature of the end user.	8
Figure 2.3	Ansible Standalone Management [20]	11
Figure 2.4	Chef Master/Agent management [20]	12
Figure 2.5	Orchestration vs. Software Management [31]	14
Figure 2.6	Implementing CloudFormation or HEAT [1])	15
Figure 2.7	Modifying an existing stack [1]	15
Figure 3.1	AWS Dashboard	18
Figure 3.2	OpenStack Horizon Dashboard	19
Figure 3.3	Cloud Research Automation for Infrastructure and Software (RAINS) in- terface	20
Figure 3.4	Code Snippet to create a tenant in OpenStack	23
Figure 3.5	Swim lane chart of instance creation	23
Figure 4.1	Code Snippet enable SSL encryption in Apache2	30

Chapter 1: INTRODUCTION

The "cloud" and cloud services began in 1999 with the arrival of salesforce.com delivering applications over the internet to consumers, later to be known as Software as a Service (SaaS) [6]. Shortly thereafter, Amazon released Amazon Web Services (AWS) in 2002 and later in 2006 [11] Elastic Cloud Compute (EC2) and since then the "cloud" has become a widely known word and a widely used tool, regardless of whether users are aware of the tool or not. However, the cloud is not without its downfalls. The cloud can be very difficult to set up and often requires teams of individuals in a company to ensure smooth operation, including but not limited to proper up- and down-scaling based on user needs, as well as proper software configurations in line with the company's demands.

According to a white paper by Vision Solutions, 59% of Fortune 500 companies experienced a minimum of 1.6 hours of downtime per week [25]. This means that for a company who has 10,000 employees who on average make a salary of \$30 per hour [17], or \$60,000 per year, this downtime can potentially create loss of \$480,000 weekly or nearly 25 million dollars annually, not including the cost of benefits, loss of sales, reputation impact from services being unavailable or other negative impacts towards the company. The cloud is capable of helping alleviate some of this downtime and, unlike traditional hardware, can be reallocated for larger or smaller systems or clusters as needed.

Unfortunately, for much of academia the cloud is still in its infancy and is rarely utilized to its potential. This slower movement towards the cloud in academia can be attributed to faculty lacking the full scope of work, support, as well as the time and resources that companies are able to devote to integrating and efficiently using the cloud [7]. Frequently in academia, research drives faculty to look at new methods to solve problems. These new methods, computationally, may require additional computers to build a larger cluster to compute a problem. Acquiring hardware can be the hardest problem for faculty with limitations on funding and space. In addition, faculty members have other requirements imposed upon them like teaching potentially up to four

courses per semester [9]. Likewise, for tenured and tenure track faculty, service to the university and community at large takes away valuable time from them to be able to try new methods such as implementing the cloud into their research endeavors [2]. Similarly, a university's information technology department frequently lacks the expertise and resources to be able to sustain and support the cloud for the diverse research and computational needs of a university.

Potential solutions exist to alleviate issues related to the lack of infrastructure and deployment and management of software. Utilizing services offered by commercial products such as Amazon's AWS, Rackspace Private Cloud, Microsoft Azure and countless others can solve the infrastructure issue. Similarly, all of these companies and others offer SaaS (Software as a Service), capable of deploying software ranging from static items like firewalls to dynamic software like machine learning software.

These companies can allow beginners to quickly and easily do machine learning on data and have results in a short period of time. However, the main issue that can arise with using these types of commercial services is that there is a degree of vendor lock-in. Vendor lock-in happens when users use that company's specific tools where they are unable to see the code running under the covers because it is proprietary. Not only does this limit the users understanding of the underlying processes, but it also creates additional costs and barriers when trying to switch to a different cloud provider as certain tools may not be one-to-one compatible with other tools offered by the competition. The direct and indirect costs associated with this kind of set-up could be a major barrier for implementing the cloud at academic institutions. Another example of vendor lock-in is that Amazon for example will not charge anything for all data being transferred into Amazon S3 or Glacier, their short and long term storage options. However, to transfer any significant amount of data (above 1GB/month) out of Amazon, it will cost 9 cents per GB. For a data analytics platform, this could easily add up especially if you are a long term user.

OpenStack, on the other hand, is an open source alternative to provide infrastructure, which allows the user to manipulate the virtual machines and any software without creating vendor lock-in due to its open source nature. Specifically, OpenStack allows users to better utilize their own