

# Cloud Automation with Configuration Management using CHEF Tool

Ramandeep Singh  
USICT, GGSIPU

**Abstract:-** Automated, efficient software deployment is essential for today's modern cloud hosting providers. With advances in cloud technology, on demand cloud services offered by public providers are becoming gradually powerful, anchoring the ecosystem of cloud services. Moreover, the DevOps teams are in much bigger focus now since they are responsible for the automation and provisioning of the whole environment along with the client application. This paper focuses upon the automation of customer application right from environment provisioning to application deployment.

According to a white paper by Vision Solutions, 59% of Fortune 500 companies experienced a minimum of 1.6 hours of downtime per week. This means that for a company who has 10,000 employees who on average make a salary of \$30 per hour, or \$60,000 per year, this downtime can potentially create a loss of \$480,000 weekly or nearly 25 million dollars annually, not including the cost of benefits, loss of sales, or negative impact to the reputation of the provider from services being unavailable. Therefore, it is of the utmost importance for a company's servers to have their services installed, configured, and running as quickly as possible and as consistent as possible to help reduce costs. This translates into automated deployment and configuration.

**Keywords:-** Cloud computing, Cloud Service Models, Cloud delivery models, Configuration Management, automation, Chef, Puppet, Ansible.

## 1. INTRODUCTION

(Cloud and Configuration Management)

### 1.1 Overview

Cloud computing is a model for enabling global, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [7]. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

### 1.2 What is Cloud Computing?

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [8]. This cloud model is composed of five essential characteristics, three service models, and four deployment models.

For a service to be considered a *cloud service*, it must have the following "Essential features:

On-demand self-service  
Broad network access  
Resource pooling  
Rapid elasticity and Measured service

### 1.3 Introduction to Cloud Service Models

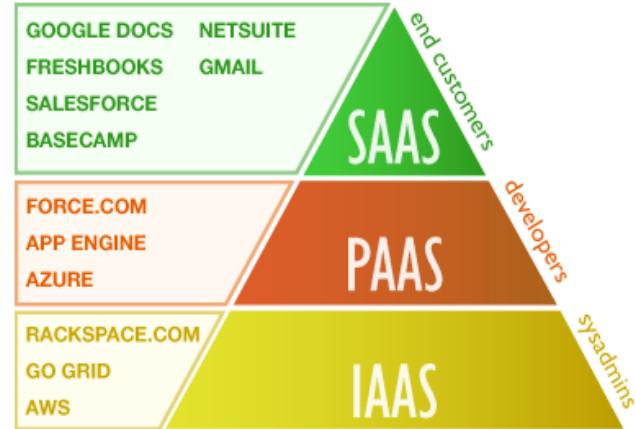


Fig 1: Types of Cloud Services Models [7]

**1.4 Types of Cloud/ Introduction to cloud delivery models**  
Cloud delivery models refer to how a cloud solution is used by an organization, where the data is located, and who operates the cloud solution [2]. Cloud computing supports multiple delivery models that can deliver the capabilities needed in a cloud solution.

The cloud delivery models are as follows:

- Public cloud
- Private cloud
- Hybrid cloud
- Community cloud

Table 1: Types of Cloud Delivery Models [8]

#### (1.4.1) Public clouds

A public cloud is one in which the cloud infrastructure is made available to the general public or a large industry group over the Internet. The infrastructure is not owned by the user, but by an organization that provides cloud services. Services can be provided either at no cost, as a subscription, or as a pay-as-you-go model.

Examples of public clouds include IBM SoftLayer, Amazon Elastic Compute Cloud (EC2), Google AppEngine, and Microsoft Azure App Service [8].

#### (1.4.2) Private clouds

A private cloud refers to a cloud solution where the infrastructure is provisioned for the limited use of a single

organization. The organization often acts as a cloud service provider to internal business units that obtain all the benefits of a cloud without having to provision their own infrastructure. By consolidating and centralizing services into a cloud, the organization benefits from centralized service management and economies of scale.

A private cloud provides an organization with some advantages over a public cloud. The organization gains greater control over the resources that make up the cloud. In addition, private clouds are ideal when the type of work being done is not practical for a public cloud because of network latency, security, or regulatory concerns.

A private cloud can be owned, managed, and operated by the organization, a third party, or a combination. The private cloud infrastructure is usually provisioned on the organization's premises, but it can also be hosted in a data centre that is owned by a third party.

#### (1.4.3) Community clouds

A community cloud shares the cloud infrastructure across several organizations in support of a specific community that has common concerns (for example, mission, security requirements, policy, and compliance considerations). The primary goal of a community cloud is to have participating organizations realize the benefits of a public cloud, such as shared infrastructure costs and a pay-as-you-go billing structure, with the added level of privacy, security, and policy compliance that is usually associated with a private cloud [8].

#### (1.4.4) Hybrid clouds

A hybrid cloud is a combination of various cloud types (public, private, and community). Each cloud in the hybrid mix remains a unique entity, but is bound to the mix by technology that enables data and application portability. The hybrid approach allows a business to take advantage of the scalability and cost-effectiveness of off-premise third-party resources without revealing applications and data beyond the corporate intranet [8]. A well-constructed hybrid cloud can service secure, mission-critical processes, such as receiving customer payments (a private cloud service), and secondary processes such as employee payroll processing (a public cloud service).

The challenge for a hybrid cloud is the difficulty in effectively creating and governing such a solution. Services from various sources must be obtained and provisioned as though they originated from a single location, and interactions between on-premises and off-premises components make the employment even more complicated.

#### Lots of Cloud Providers!

	SaaS	PaaS	IaaS
Private Cloud	CiscoWebex, Citrix GoToMeeting	Apprenda, Stackato	VMware, Hyper-V, Openstack, CloudStack
Public Cloud	Google Apps, Salesforce, Cloud9 Analytics, Office 365	Google Apps Engine, Microsoft Azure, VMware	Amazon EC2, Rackspace, Google Compute Engine (GCE)
Hybrid Cloud		Cloud Foundry, Custom	Cisco Intercloud Fabric, Custom



Table 2: Few Cloud Providers[9]

## 2. CLOUD AND CONFIGURATION MANAGEMENT

### 2.1 How cloud is related to Configuration Management?

Cloud and cloud services have become more popular in recent years. In the cloud hosting industry, companies have found that costs can be reduced by improving up-time in servers and creating a scalable server based on load. According to a white paper by Vision Solutions, 59% of Fortune 500 companies experienced a minimum of 1.6 hours of downtime per week. This means that for a company who has 10,000 employees who on average make a salary of \$30 per hour, or \$60,000 per year, this downtime can potentially create a loss of \$480,000 weekly or nearly 25 million dollars annually, not including the cost of benefits, loss of sales, or negative impact to the reputation of the provider from services being unavailable. Consequently, it is of the utmost importance for a company's servers to have their services installed, configured, and running as quickly as possible and as consistent as possible to help reduce costs. This translates into automated deployment and configuration.

### 2.2 What is Configuration Management?

Configuration management (CM) is a systems engineering process for establishing and maintaining consistency of a product's performance, efficient, and physical attributes with its requirements, design, and operational information throughout its life [2].

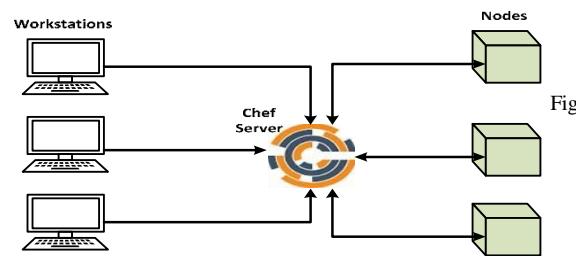
Configuration management can be used to maintain OS configuration files.

Various tools for Configuration Management include Ansible, Bcfg2, CFEngine, Chef, Otter, Puppet, SaltStack etc [2].

## 3. AUTOMATION TOOLS

### 3.1 Chef

This is a system and cloud infrastructure automation tool for installing applications and software to bare metal, virtual machine, and container-based clouds [1]. The configuration is written in Ruby DSL, and it uses the concepts of organizations, environments, cookbooks, recipes and resources, all driven by supplied or derived attributes. The tool has a set of control parts that work together to provide its functionality. Chef Workstation is used to control the deployment of configurations from the Chef Server to Chef managed nodes. Nodes are bootstrapped with agents and pull configurations from the server. The core is developed in Erlang and is designed to provide scale to tens of thousands of servers [1]. It is developed around an infrastructure-as-a-code model with version control integral to the workstation. Directives run top to bottom, and the cookbooks, does not matter how many times are running, generate the same result.



2: Architecture of Chef [1]

There are three main components:

- Chef Server is the most important part because it stores the whole configuration data for all nodes. It has also the role to administrate the access rights.
- Chef Workstation is the place where the cookbooks, recipes and all of the configuration parts are created (to be deployed to Chef Nodes through Chef Server).
- Chef Nodes are the places where the cookbooks, recipes and all of the configuration parts are stored.

#### Advantages

- Large community of cookbooks and development tools.

- Ability to handle physical, virtual, and containers deployments.
- Provides hosted services.
- Excellent at managing operating system.

#### Disadvantages

- Complex to set up the entire stuff because it requires good understanding of Ruby.
- Huge amount of documentation.
- It requires an agent to be installed and pulled configuration

## 4. COMPARISON OF THE TOOLS

### 4.1 Comparing various tools on the basis of pros and cons

Tool	Pros	Cons
Ansible	<ul style="list-style-type: none"> <li>• Easy to start, install, deploy and manage.</li> <li>• Highly accessible architecture</li> <li>• Easy learning curve thanks to the use of YAML.</li> </ul>	<ul style="list-style-type: none"> <li>• No consistency between formats of input, output, and config files</li> <li>• Fights with performance speed at times.</li> <li>• No consistency between formats of input, output, and config files.</li> </ul>
Puppet	<ul style="list-style-type: none"> <li>• Automation of compliance across the environment, high value to enterprise</li> <li>• Web UI &amp; Reporting tools</li> </ul>	<ul style="list-style-type: none"> <li>• Hard to learn for new users</li> <li>• Difficult to scale</li> <li>• Model-driven approach means less control compared to code driven approaches.</li> </ul>
Chef	<ul style="list-style-type: none"> <li>• Large community of cookbooks and development tools.</li> <li>• Ability to handle physical, virtual, and containers deployment</li> <li>• Provides hosted services like Chef</li> <li>• Excellent at managing operating systems</li> </ul>	<ul style="list-style-type: none"> <li>• Complex to set up the whole stuff because it requires good considerate of Ruby.</li> <li>• Huge amount of documentation.</li> <li>• It's not a simple tool, which can lead to large code bases and complicated environments.</li> <li>• It requires an agent to be installed and pulled configuration</li> </ul>

Table 3: Pros and cons of the tools

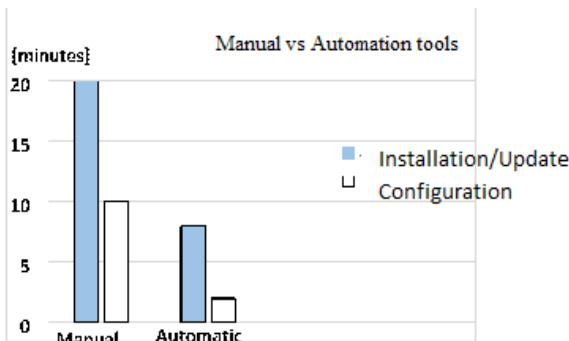
## 5. CONCLUSION

The manual installation, updation and configuration of software packages requires a lot of manpower and time. We have compared the manual installation with automation tools in order to provide an accurate cloud deployment. The solution can be easily scaled by cloning the cookbook/playbook or module for the Compute node and by increment assigning of IP addresses. The automation tools decrease the time required to complete the tasks mentioned from a few hours to few minutes.

### 4.2 Comparison of the tools on the basis of their working

Characteristic	Ansible	Chef	Puppet
<u>Language</u>	Python	Ruby	Ruby
<u>License</u>	GPL	Apache	Apache
<u>Authorization</u>	Yes - SSH	Yes-Keys	Yes
<u>Encrypts</u>	Yes – SSH	Yes-SSL	Yes-SSL
<u>Mechanism</u>	Push	Pull	Pull
<u>Agent-less</u>	Yes	No	No
<u>GUI</u>	Yes	Yes	Yes

Table 4: Comparison of tools on the basis of their characteristics



Performance evaluation of automation tools vs. manual

Fig 5: Performance Evaluation of Automation tools v/s doing it manually  
[1]

## 6. ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my mentor Dr. Ravindra Kumar Purwar, Associate Professor of USICT, who gave me an opportunity to do a research and review on the topic “Cloud Automation with Configuration Management”, which helped me in doing my work and prepared myself with the vast knowledge of the related area. I am thankful to him.

## 7. REFERENCES

- [1] Eduard Luchian, Cosmin Filip, Andrei Bogdan Rus, Iustin-Alexandru Ivanciu, Virgil Dobrota, Automation of the Infrastructure and Services for an OpenStack Deployment Using Chef Tool, IEEE International Conference on Cloud Engineering IC2E, 2017, pp. 295-302.
- [2] Dmitry Duplyakin and Robert Ricci, Introducing Configuration Management Capabilities into CloudLab Experiments, IEEE INFOCOM International Workshop on Computer and Networking Experimental Research Using Testbeds, 2016, pp. 453-458
- [3] M. Boschetti and P. Ruiu, A Cloud automation platform for flexibility in applications and resources provisioning, 9th International Conference on Complex, Intelligent, and Software Intensive Systems, 2015, pp.204-208
- [4] Gregory Katsaros, Alexander Lenk, Michael Menzel, Jannis Rake, Ryan Skipp, Jacob Eberhardt, Cloud application portability with TOSCA, Chef and Openstack, IEEE International Conference on Cloud Engineering, 2014, pp.295-302.
- [5] Nishant Kumar Singh, Sanjeev Thakur, Himanshu Chaurasiya and Himanshu Nagdev, Automated Provisioning of Application in IAAS Cloud using Ansible Configuration Management, 1st International Conference on Next Generation Computing Technologies (NGCT-2015) Dehradun, India, 4-5 September 2015, pp. 81-85.
- [6] Johannes Hintsch, Carsten Görling, and Klaus Turowski, Modularization of Software as a Service Products: A Case Study of the Configuration Management Tool Puppet, Third International Conference on Enterprise Systems, 2015, pp. 184-191.
- [7] <http://www.globaldots.com/cloud-computing-types-of-cloud/>
- [8] <http://www.redbooks.ibm.com/redpapers/pdfs/redp4873.pdf>
- [9] <https://CiscoDevNet/devnet-1008-private-public-or-hybrid-cloud-which-cloud-should-i-choose>