International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 04, April-2019

# Cloud Automation with Configuration Managementusing CHEF Tool

Ramandeep Singh, Dr. Ravindra Kumar Purwar
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*

DevOps is a blend of two terms – development and operations. It is considered to be the most effective way to foster collaboration and eliminate the walls of confusion that exist between software developers and operations teams, by way of shared experiences and suggested solutions [2].

DevOps has been identified as a phenomenon whereby stakeholders of a software development team work together to deliver software continuously, allowing the business to seize emerging and existing market opportunities while reducing the
time required for inclusion of client feedback. However, little has been done to develop adoption strategies/frameworks for this phenomenon. The absence of such strategies may result in DevOps not being adequately communicated and its impact not fully comprehended in both the practitioner and academic research communities. This study investigates the factors that are hindering the adoption of DevOps and proposes strategies to address them using both literature study and interviews with practitioners actively involved in the DevOps movement[2]. A proposed conceptual framework has been developed both to strategize the adoption of DevOps, and to contribute input for future research.

DevOps is one of the most popular approaches for software delivery nowadays. Even though there is no unified definition of DevOps, there is wide consensus about the set of practices that are part of it[1]. Two of those practices are Infrastructure as Code and Continuous Delivery, which bring new artifacts into the Software Development lifecycle. These new artifacts have direct impact on Software Configuration Management, which is a well-known practice that has been around for decades in the Software Engineering discipline. In particular, these new practices have a direct impact on Version Control. This article describes a Version Control Strategy to manage these new artifacts introduced by DevOps initiatives. The proposed strategy covers the identification of artifacts, versioning tools,version naming conventions and traceability between different artifacts versions. The strategy was validated in three cases of real world projects where it was successfully applied. Each case corresponds to a different kind of organization and in each case a different set of tools where used. Based on these cases, benefits and possible improvements are identified.

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:

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 04, April-2019

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

### *1.3 Introduction to Cloud Service Models*

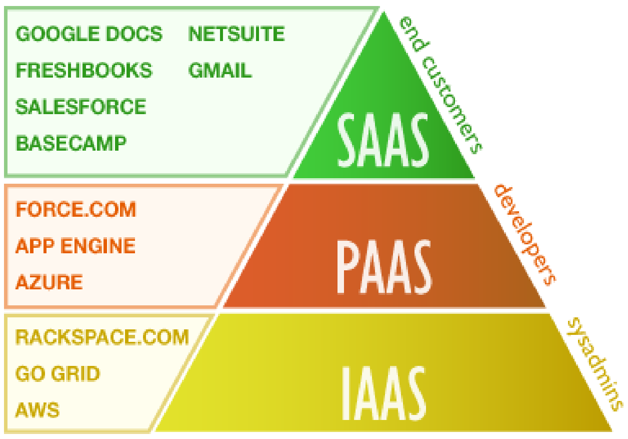


Fig 1: Types of Cloud Services Models [7]

### *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*

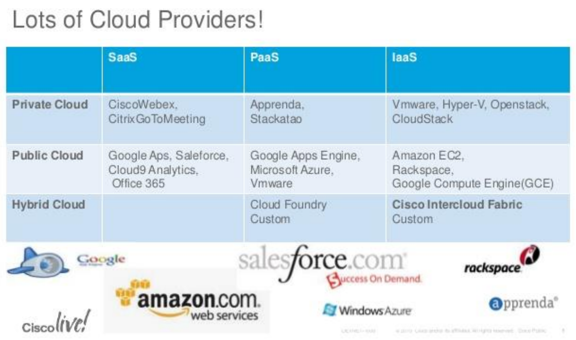
A private cloud refers to a cloud solution where the infrastructure is provisioned for the limited use of a single organization. The organization often acts as a cloud service provider to internal business units that obtain all the benefits of a cloud without having to provision their own infrastructure. 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-premise components make the employment even more complicated.


Lots of Cloud Providers!

| | SaaS | PaaS | IaaS |
|---|---|---|---|
| **Private Cloud** | CiscoWebex, CitrixGoToMeeting | Apprenda, Stackatao | Vmware, Hyper-V, Openstack, CloudStack |
| **Public Cloud** | Google Aps, Saleforce, 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.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

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.

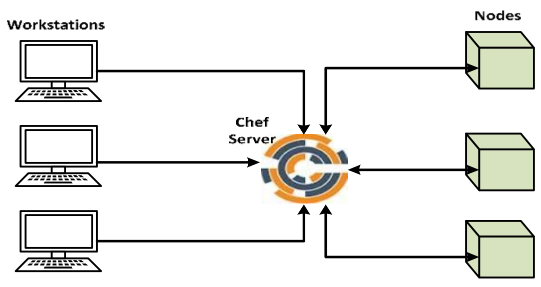


Fig 3: Architecture of Chef [1]

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 were 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*

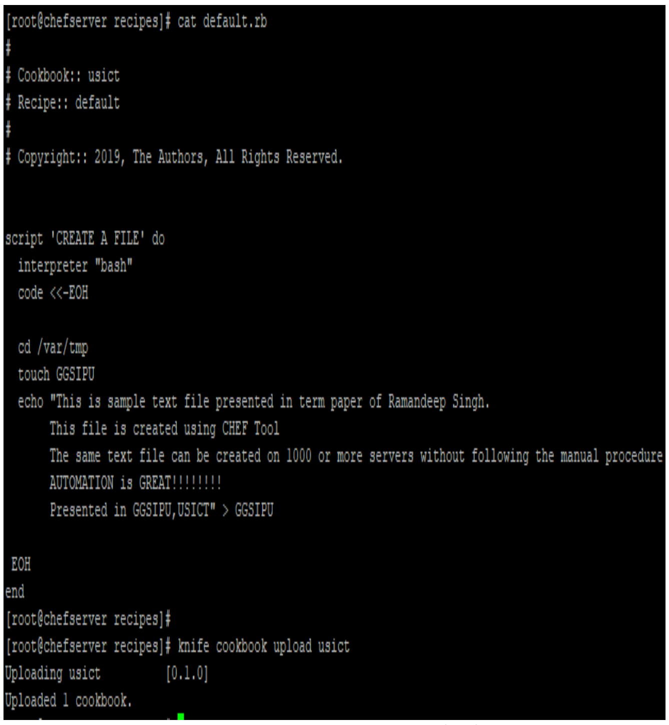
- Complex to set up the entire stuff because it requires good understanding of Ruby.
- Huge amount of documentation.
- It requires an agent to be installed and pulled configuration

## 4. IMPLEMENTATION

### *4.1 Creating/Copying a file on multiple servers*

A number of times, we have a requirement to copy the exact same file on hundreds of servers. Its practical to create a file on a couple of servers but creating a file with exact same content on hundreds of servers is cumbersome and waste of manpower. Hence, with tools like CHEF, we can create a file using the defined resources and have the file deployed on any number of servers.
In the figures below, we create a file by writing a small script in the cookbook. Also, for user creation, use the resource "user" to add users, update existing users, remove users, and to lock/unlock user passwords. Moreover, Use the package resource to manage packages. This resource is the base resource for several other resources used for package management on specific platforms. While it is possible to use each of these specific resources, it is recommended to use the package resource as often as possible. All we need to do is to upload the cookbook to the server which is linked to the clients (on which configuration changes have to be deployed).

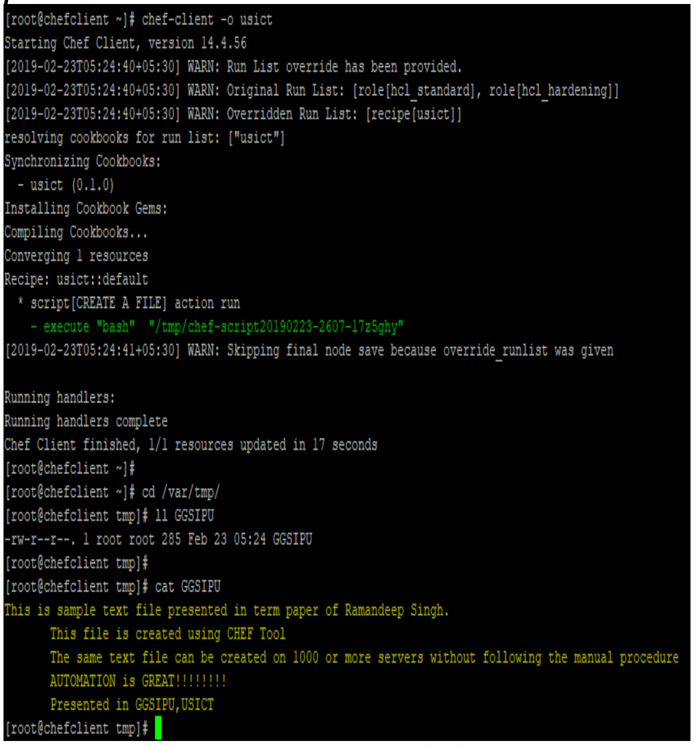
(This work is licensed under a Creative Commons Attribution 4.0 International License.)



```
[root@chefserver recipes]# cat default.rb
#
# Cookbook:: usict
# Recipe:: default
#
# Copyright:: 2019, The Authors, All Rights Reserved.


script 'CREATE A FILE' do
  interpreter "bash"
  code <<-EOH

  cd /var/tmp
  touch GGSIPU
  echo "This is sample text file presented in term paper of Ramandeep Singh.
       This file is created using CHEF Tool
       The same text file can be created on 1000 or more servers without following the manual procedure
       AUTOMATION is GREAT!!!!!!!!
       Presented in GGSIPU,USICT" > GGSIPU

 EOH
end
[root@chefserver recipes]#
[root@chefserver recipes]# knife cookbook upload usict
Uploading usict         [0.1.0]
Uploaded 1 cookbook.
```

Fig 4: Creating and uploading a cookbook for file creation

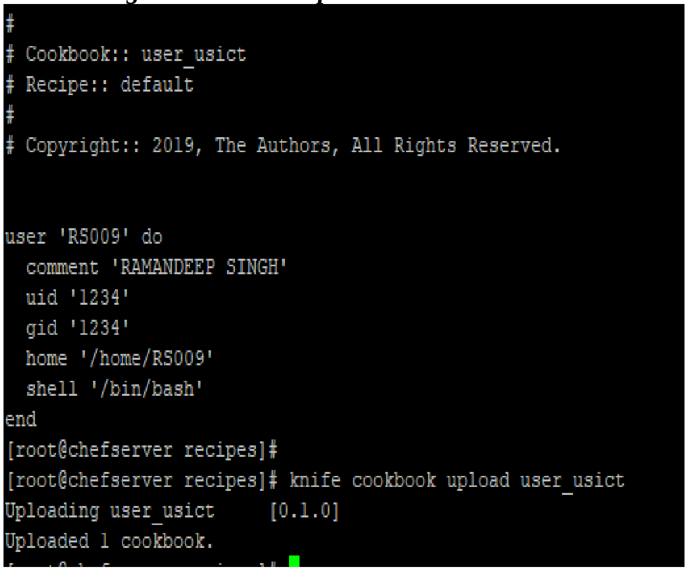
### Executing a cookbook on client and Display the data of file



```
[root@chefclient ~]# chef-client -o usict
Starting Chef Client, version 14.4.56
[2019-02-23T05:24:40+05:30] WARN: Run List override has been provided.
[2019-02-23T05:24:40+05:30] WARN: Original Run List: [role[hcl_standard], role[hcl_hardening]]
[2019-02-23T05:24:40+05:30] WARN: Overridden Run List: [recipe[usict]]
resolving cookbooks for run list: ["usict"]
Synchronizing Cookbooks:
  - usict (0.1.0)
Installing Cookbook Gems:
Compiling Cookbooks...
Converging 1 resources
Recipe: usict::default
  * script[CREATE A FILE] action run
    - execute "bash"  "/tmp/chef-script20190223-2607-17z5ghy"
[2019-02-23T05:24:41+05:30] WARN: Skipping final node save because override_runlist was given

Running handlers:
Running handlers complete
Chef Client finished, 1/1 resources updated in 17 seconds
[root@chefclient ~]#
[root@chefclient ~]# cd /var/tmp/
[root@chefclient tmp]# ll GGSIPU
-rw-r--r--. 1 root root 285 Feb 23 05:24 GGSIPU
[root@chefclient tmp]#
[root@chefclient tmp]# cat GGSIPU
This is sample text file presented in term paper of Ramandeep Singh.
       This file is created using CHEF Tool
       The same text file can be created on 1000 or more servers without following the manual procedure
       AUTOMATION is GREAT!!!!!!!!
       Presented in GGSIPU,USICT
[root@chefclient tmp]#
```

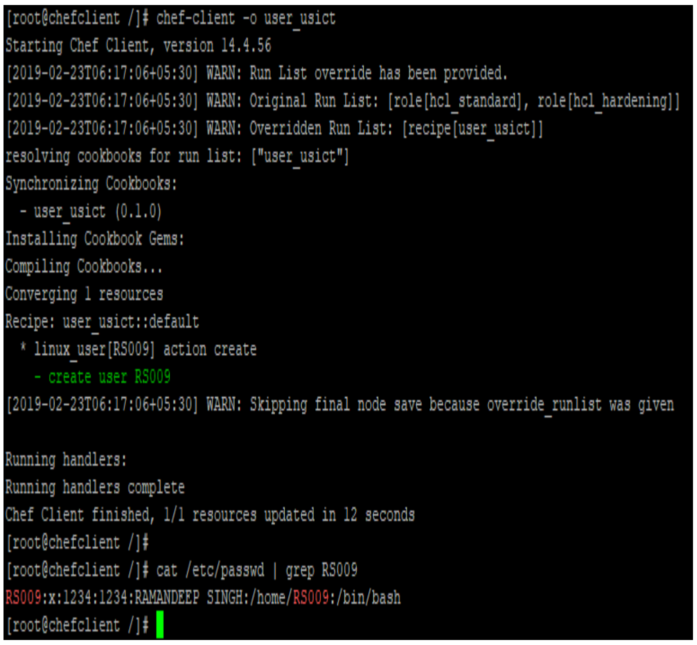
Fig 5: Executing a cookbook on client for file creation

### 4.2 Creating a User on multiple servers



```
#
# Cookbook:: user_usict
# Recipe:: default
#
# Copyright:: 2019, The Authors, All Rights Reserved.


user 'RS009' do
  comment 'RAMANDEEP SINGH'
  uid '1234'
  gid '1234'
  home '/home/RS009'
  shell '/bin/bash'
end
[root@chefserver recipes]#
[root@chefserver recipes]# knife cookbook upload user_usict
Uploading user_usict     [0.1.0]
Uploaded 1 cookbook.
```

Fig 6: Creating and uploading a cookbook for user creation



```
[root@chefclient /]# chef-client -o user_usict
Starting Chef Client, version 14.4.56
[2019-02-23T06:17:06+05:30] WARN: Run List override has been provided.
[2019-02-23T06:17:06+05:30] WARN: Original Run List: [role[hcl_standard], role[hcl_hardening]]
[2019-02-23T06:17:06+05:30] WARN: Overridden Run List: [recipe[user_usict]]
resolving cookbooks for run list: ["user_usict"]
Synchronizing Cookbooks:
  - user_usict (0.1.0)
Installing Cookbook Gems:
Compiling Cookbooks...
Converging 1 resources
Recipe: user_usict::default
  * linux_user[RS009] action create
    - create user RS009
[2019-02-23T06:17:06+05:30] WARN: Skipping final node save because override_runlist was given

Running handlers:
Running handlers complete
Chef Client finished, 1/1 resources updated in 12 seconds
[root@chefclient /]#
[root@chefclient /]# cat /etc/passwd | grep RS009
RS009:x:1234:1234:RAMANDEEP SINGH:/home/RS009:/bin/bash
[root@chefclient /]#
```

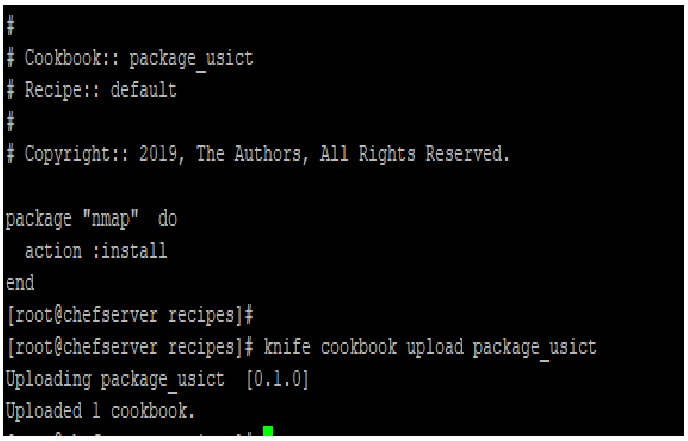
Fig 7: Executing a cookbook on client for user configuration

### 4.3 Installation of software packages on multiple servers

All of us have installed a number of software packages in our daily life. In organizational servers, there are a few packages which have to be present/install as basic packages on almost all the servers. Such packages if installed individually on each server would take a lot of time. Hence, configuration tools like CHEF reduces the time and effort to a considerable amount.

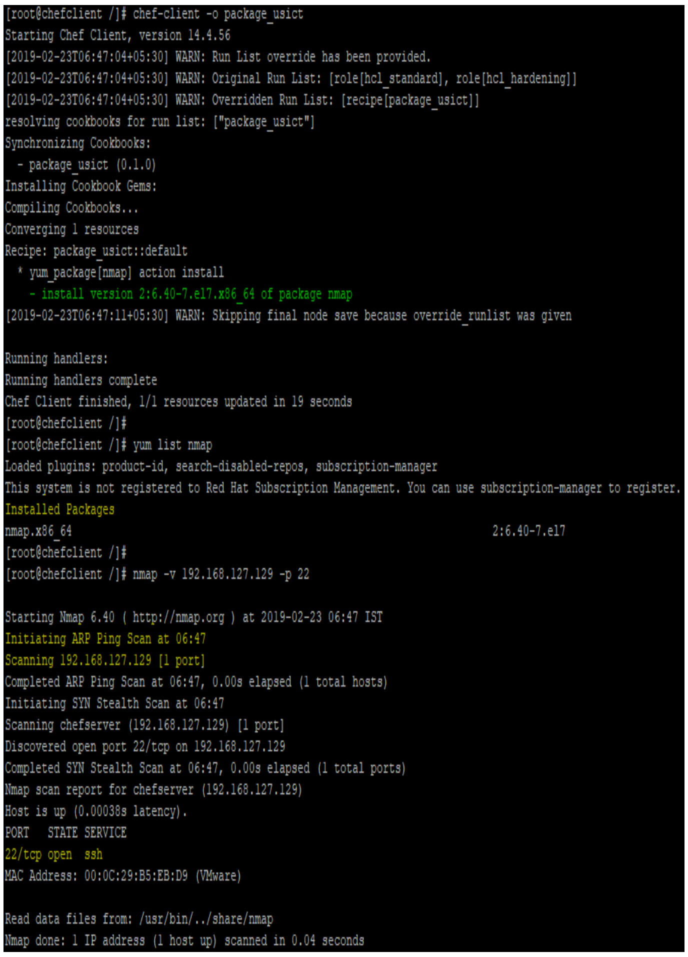
Let's install nmap package which is used for port scanning purpose.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)



```
#
# Cookbook:: package_usict
# Recipe:: default
#
# Copyright:: 2019, The Authors, All Rights Reserved.

package "nmap"  do
  action :install
end
[root@chefserver recipes]#
[root@chefserver recipes]# knife cookbook upload package_usict
Uploading package_usict   [0.1.0]
Uploaded 1 cookbook.
```

Fig 8: Installing a software package-nmap



```
[root@chefclient /]# chef-client -o package_usict
Starting Chef Client, version 14.4.56
[2019-02-23T06:47:04+05:30] WARN: Run List override has been provided.
[2019-02-23T06:47:04+05:30] WARN: Original Run List: [role[hcl_standard], role[hcl_hardening]]
[2019-02-23T06:47:04+05:30] WARN: Overridden Run List: [recipe[package_usict]]
resolving cookbooks for run list: ["package_usict"]
Synchronizing Cookbooks:
  - package_usict (0.1.0)
Installing Cookbook Gems:
Compiling Cookbooks...
Converging 1 resources
Recipe: package_usict::default
  * yum_package[nmap] action install
    - install version 2:6.40-7.el7.x86_64 of package nmap
[2019-02-23T06:47:11+05:30] WARN: Skipping final node save because override_runlist was given

Running handlers:
Running handlers complete
Chef Client finished, 1/1 resources updated in 19 seconds
[root@chefclient /]#
[root@chefclient /]# yum list nmap
Loaded plugins: product-id, search-disabled-repos, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to register.
Installed Packages
nmap.x86_64                                                                  2:6.40-7.el7
[root@chefclient /]#
[root@chefclient /]# nmap -v 192.168.127.129 -p 22

Starting Nmap 6.40 ( http://nmap.org ) at 2019-02-23 06:47 IST
Initiating ARP Ping Scan at 06:47
Scanning 192.168.127.129 [1 port]
Completed ARP Ping Scan at 06:47, 0.00s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 06:47
Scanning chefserver (192.168.127.129) [1 port]
Discovered open port 22/tcp on 192.168.127.129
Completed SYN Stealth Scan at 06:47, 0.00s elapsed (1 total ports)
Nmap scan report for chefserver (192.168.127.129)
Host is up (0.00038s latency).
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:B5:EB:D9 (VMware)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds
```

Fig 9: Executing a cookbook to install nmap package

The package nmap is installed as we can see above. By using the mentioned command, we can use nmap for port scanning.

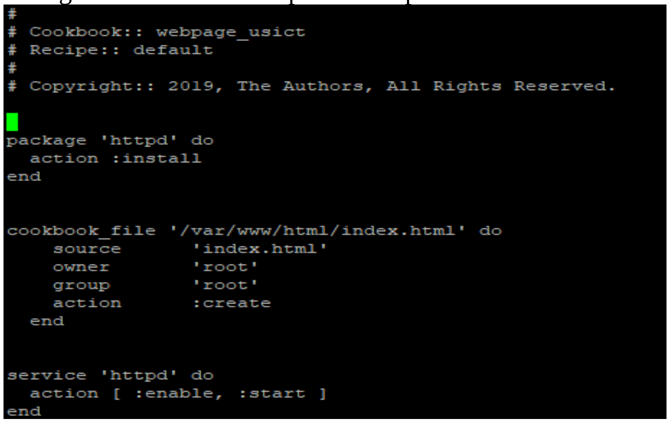
***nmap -v 192.168.127.129 -p 22, where -v means verbose and -p means port to be scanned.***

Nmap can also provide further information on targets, including reverse DNS names, operating system guesses, device types, and MAC addresses.

Hence, this is the beauty of automation and configuration tools like CHEF, they make our daily tasks much more easier and less time consuming, reduces the manpower and cost also.

### *4.4 Creating and configuring a web server/web page*

Chef can be used to install/configure a web server/web page also.You can host your own website using this.In this we are configuring Apache web server. The primary Apache configurationfile is /etc/httpd/conf/httpd.conf.

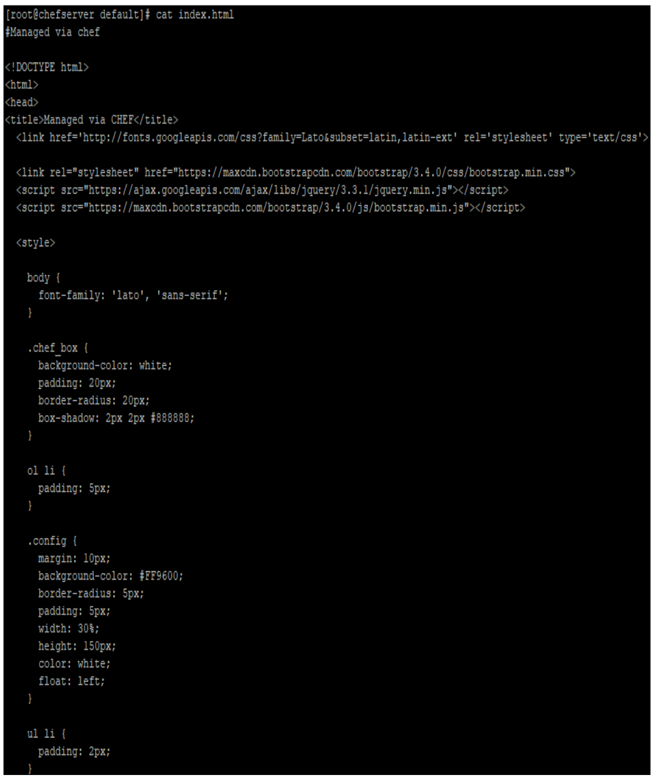


```
#
# Cookbook:: webpage_usict
# Recipe:: default
#
# Copyright:: 2019, The Authors, All Rights Reserved.

package 'httpd' do
  action :install
end


cookbook_file '/var/www/html/index.html' do
    source       'index.html'
    owner        'root'
    group        'root'
    action       :create
  end


service 'httpd' do
  action [ :enable, :start ]
end
```

Fig 10: Installing/configure apache web Server

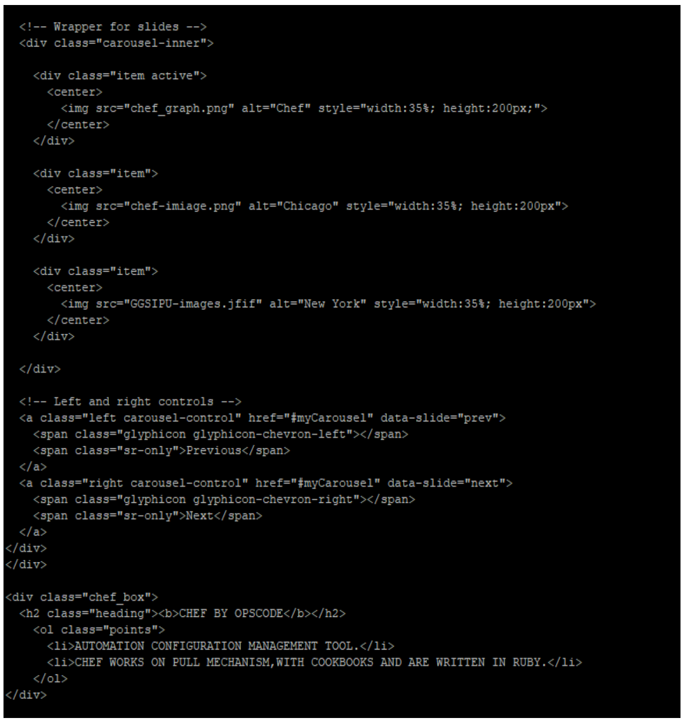
### *Create index.html file*

The index.html file is the default file a web server will serve up when you access the website. In the /var/www/html directory, create a file with the name index.html. Add the content. PFB thindex.html

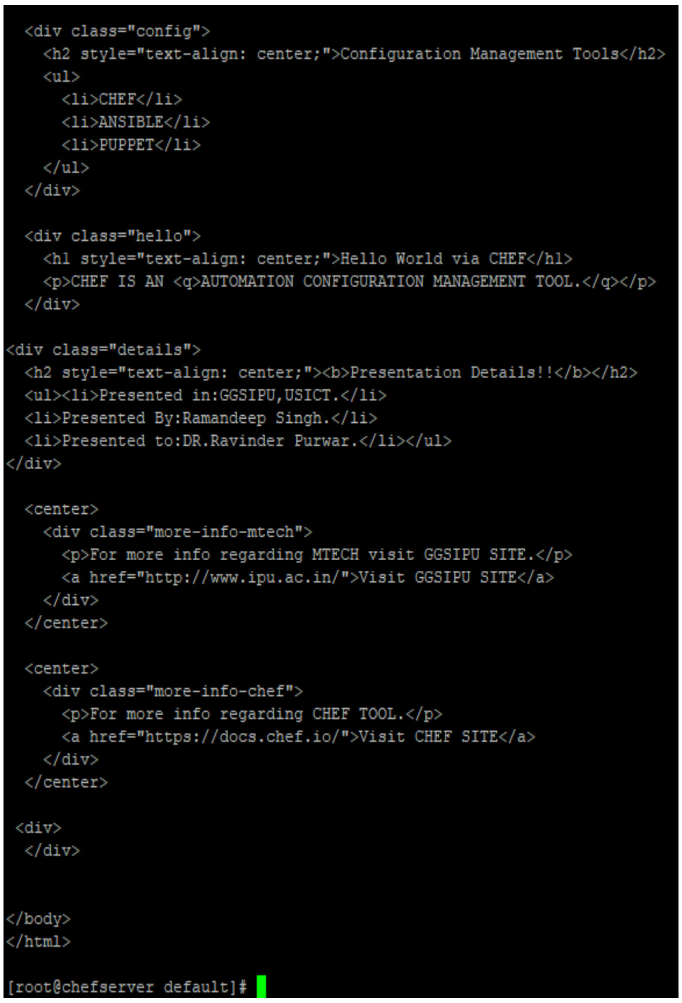


```
[root@chefserver default]# cat index.html
#Managed via chef

<!DOCTYPE html>
<html>
<head>
<title>Managed via CHEF</title>
  <link href='http://fonts.googleapis.com/css?family=Lato&subset=latin,latin-ext' rel='stylesheet' type='text/css'>

  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>

  <style>

    body {
      font-family: 'lato', 'sans-serif';
    }

    .chef_box {
      background-color: white;
      padding: 20px;
      border-radius: 20px;
      box-shadow: 2px 2px #888888;
    }

    ol li {
      padding: 5px;
    }

    .config {
      margin: 10px;
      background-color: #FF9600;
      border-radius: 5px;
      padding: 5px;
      width: 30%;
      height: 150px;
      color: white;
      float: left;
    }

    ul li {
      padding: 2px;
    }
```

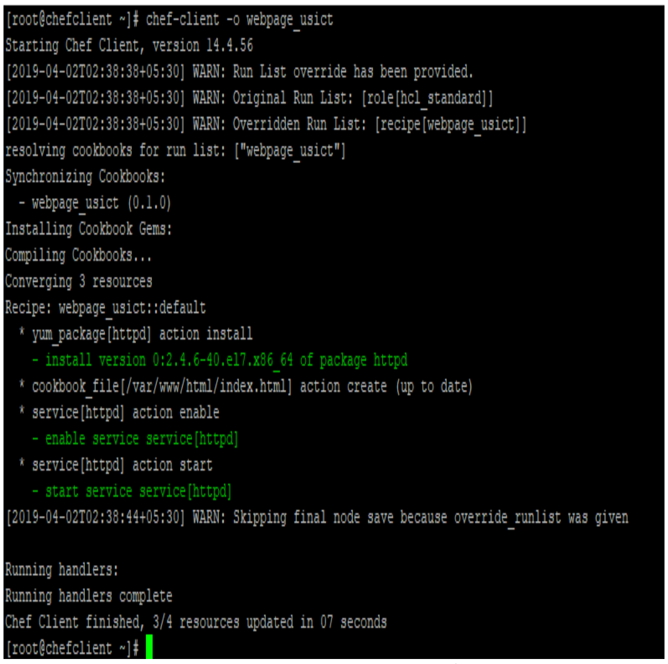
(This work is licensed under a Creative Commons Attribution 4.0 International License.)



```
  <!-- Wrapper for slides -->
  <div class="carousel-inner">

    <div class="item active">
      <center>
        <img src="chef_graph.png" alt="Chef" style="width:35%; height:200px;">
      </center>
    </div>

    <div class="item">
      <center>
        <img src="chef-imiage.png" alt="Chicago" style="width:35%; height:200px">
      </center>
    </div>

    <div class="item">
      <center>
        <img src="GGSIPU-images.jfif" alt="New York" style="width:35%; height:200px">
      </center>
    </div>

  </div>

  <!-- Left and right controls -->
  <a class="left carousel-control" href="#myCarousel" data-slide="prev">
    <span class="glyphicon glyphicon-chevron-left"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="right carousel-control" href="#myCarousel" data-slide="next">
    <span class="glyphicon glyphicon-chevron-right"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
</div>

<div class="chef_box">
  <h2 class="heading"><b>CHEF BY OPSCODE</b></h2>
    <ol class="points">
      <li>AUTOMATION CONFIGURATION MANAGEMENT TOOL.</li>
      <li>CHEF WORKS ON PULL MECHANISM,WITH COOKBOOKS AND ARE WRITTEN IN RUBY.</li>
    </ol>
</div>
```



```
  <div class="config">
    <h2 style="text-align: center;">Configuration Management Tools</h2>
    <ul>
      <li>CHEF</li>
      <li>ANSIBLE</li>
      <li>PUPPET</li>
    </ul>
  </div>

  <div class="hello">
    <h1 style="text-align: center;">Hello World via CHEF</h1>
    <p>CHEF IS AN <q>AUTOMATION CONFIGURATION MANAGEMENT TOOL.</q></p>
  </div>

<div class="details">
  <h2 style="text-align: center;"><b>Presentation Details!!</b></h2>
  <ul><li>Presented in:GGSIPU,USICT.</li>
  <li>Presented By:Ramandeep Singh.</li>
  <li>Presented to:DR.Ravinder Purwar.</li></ul>
</div>

  <center>
    <div class="more-info-mtech">
      <p>For more info regarding MTECH visit GGSIPU SITE.</p>
      <a href="http://www.ipu.ac.in/">Visit GGSIPU SITE</a>
    </div>
  </center>

  <center>
    <div class="more-info-chef">
      <p>For more info regarding CHEF TOOL.</p>
      <a href="https://docs.chef.io/">Visit CHEF SITE</a>
    </div>
  </center>

 <div>
  </div>


</body>
</html>

[root@chefserver default]#
```

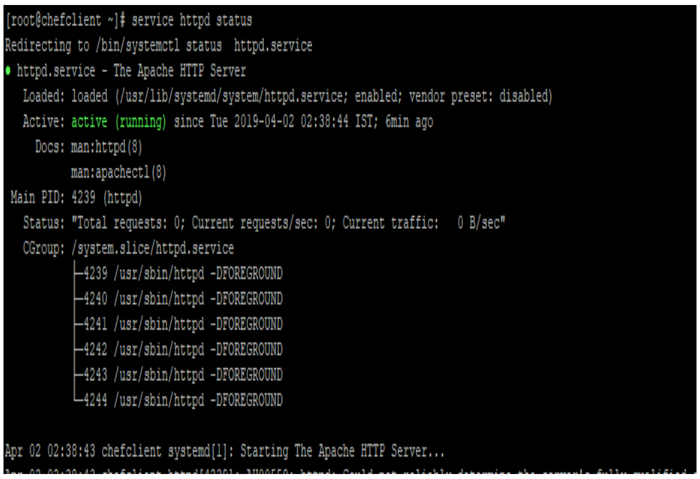


```
[root@chefclient ~]# chef-client -o webpage_usict
Starting Chef Client, version 14.4.56
[2019-04-02T02:38:38+05:30] WARN: Run List override has been provided.
[2019-04-02T02:38:38+05:30] WARN: Original Run List: [role[hcl_standard]]
[2019-04-02T02:38:38+05:30] WARN: Overridden Run List: [recipe[webpage_usict]]
resolving cookbooks for run list: ["webpage_usict"]
Synchronizing Cookbooks:
  - webpage_usict (0.1.0)
Installing Cookbook Gems:
Compiling Cookbooks...
Converging 3 resources
Recipe: webpage_usict::default
  * yum_package[httpd] action install
    - install version 0:2.4.6-40.el7.x86_64 of package httpd
  * cookbook_file[/var/www/html/index.html] action create (up to date)
  * service[httpd] action enable
    - enable service service[httpd]
  * service[httpd] action start
    - start service service[httpd]
[2019-04-02T02:38:44+05:30] WARN: Skipping final node save because override_runlist was given

Running handlers:
Running handlers complete
Chef Client finished, 3/4 resources updated in 07 seconds
[root@chefclient ~]#
```

Fig 11:Executing a cookbook to install/configure apache

Once mentioned below pre-requisites done, then check apache service:

- Package httpd should be installed
- Index.html should be there in /var/www/html
- Apache service should be running

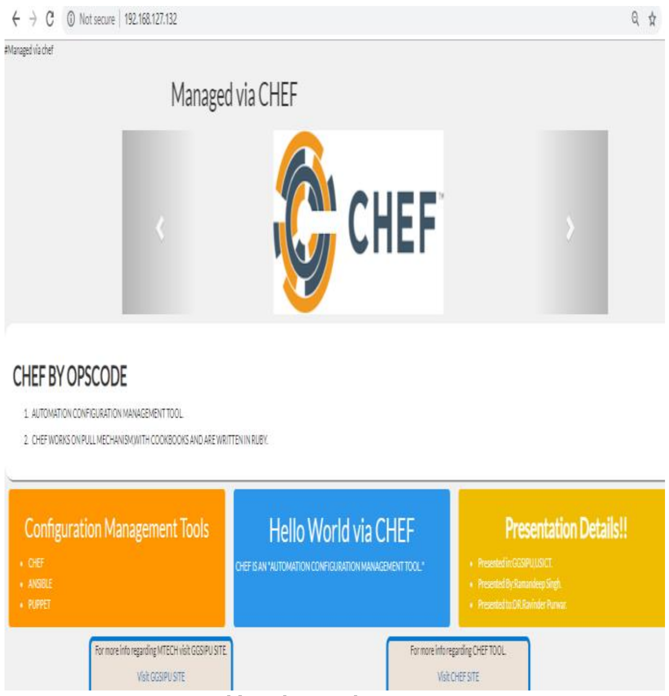


```
[root@chefclient ~]# service httpd status
Redirecting to /bin/systemctl status  httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2019-04-02 02:38:44 IST; 6min ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 4239 (httpd)
   Status: "Total requests: 0; Current requests/sec: 0; Current traffic:   0 B/sec"
   CGroup: /system.slice/httpd.service
           ├─4239 /usr/sbin/httpd -DFOREGROUND
           ├─4240 /usr/sbin/httpd -DFOREGROUND
           ├─4241 /usr/sbin/httpd -DFOREGROUND
           ├─4242 /usr/sbin/httpd -DFOREGROUND
           ├─4243 /usr/sbin/httpd -DFOREGROUND
           └─4244 /usr/sbin/httpd -DFOREGROUND

Apr 02 02:38:43 chefclient systemd[1]: Starting The Apache HTTP Server...
```

Fig 12: Checking apache service status

As apache is running we can see above, so lastly open any browser, and hit the server's ip then you can or able to see your own customizable website/webpage.You can customize your index.html as per your requirement and then can automate this process by using CHEF tool.
This is how you can create your own customizable website/webpage easily.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)



Not secure | 192.168.127.132

#Managed via chef

Managed via CHEF

CHEF BY OPSCODE

1. AUTOMATION CONFIGURATION MANAGEMENT TOOL.
2. CHEF WORKS ON PULL MECHANISM,WITH COOKBOOKS AND ARE WRITTEN IN RUBY.

Configuration Management Tools
- CHEF
- ANSIBLE
- PUPPET

Hello World via CHEF

CHEF IS AN "AUTOMATION CONFIGURATION MANAGEMENT TOOL."

Presentation Details!!
- Presented in:GGSIPU,USICT.
- Presented By:Ramandeep Singh.
- Presented to:DR.Ravinder Purwar.

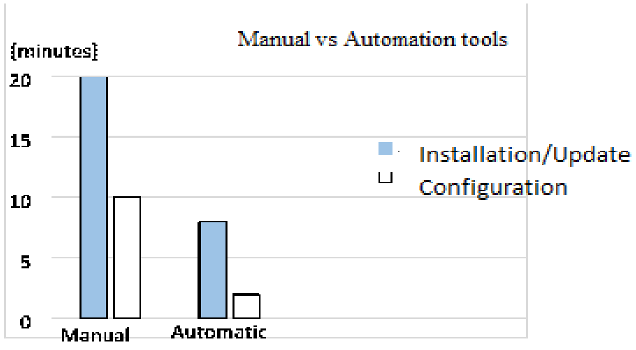
For more info regarding MTECH visit GGSIPU SITE.
Visit GGSIPU SITE

For more info regarding CHEF TOOL.
Visit CHEF SITE

Fig 13: Customizable website/webpage using CHEF TOOL

## 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 a 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.



Performance evaluation of automation tools vs. manual

Fig 14: 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] Nicolás Paez, Versioning Strategy for DevOps Implementations, Department of Science and Technology Universidad Nacional de Tres de Febrero, IEEE 2018, pp. 1-6.
[2] Morgan B. Kamuto, Josef J. Langerman, Factors Inhibiting the Adoption of DevOps in Large Organisations: South African Context, 2017 2nd IEEE International Conference On Recent Trends InElectronics Information & Communication Technology, pp. 48-51
[3] 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.
[4] James O. Benson, John J. Prevost, and Paul Rad, Survey of Automated Software Deployment for Computational and Engineering Research, IEEE Transactions, 2016.
[5] 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
[6] 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
[7] 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.
[8] 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.
[9] Johannes Hintsch, Carsten G¨orling, 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
[10] http://www.globaldots.com/cloud-computing-types-of-cloud/
[11] http://www.redbooks.ibm.com/redpapers/pdfs/redp4873.pdf
[12] https://CiscoDevNet/devnet-1008-private-public-or-hybrid-cloud-which-cloud-should-i-choose

(This work is licensed under a Creative Commons Attribution 4.0 International License.)