Volume: 10 Issue: III Month of publication: March 2022
DOI: https://doi.org/10.22214/ijraset.2022.40909

Chef Automation on Google Cloud

N. Kumaran[1], Ramya H[2], V. Apoorva[3]
[1]Assistant Professor, Department of Computer Science and Engineering
[2, 3]B.Tech, Department of Information Technology
[1, 2, 3]Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya University, Enathur, Kanchipuram, Tamilnadu

Abstract: This project is about Chef Automation. Generally if you want to install particular software or a package on a single server you can install it easily. Imagine you are maintaining an organization and you have thousands of servers running in your production environment. At this point, you cannot go and install packages or software's on all thousand servers and maintaining all these servers is also a difficult job. With our project which is chef automation, you can automate all these processes or any installations on all thousand servers in a single click. Here we can save a lot of time and manpower by deploying the code to all servers. In chef automation we have three nodes: Server, Workstation and Client. Chef automation is nothing but automating the processes on client machines. First, we will be writing cookbooks on workstation (cookbooks are the actual programs that need to be run on client machines). These cookbooks are written in ruby programming language. After developing cookbooks these are needed to be pushed on to the server node.

Keywords: Chef, Cloud Shell, Compute Engine, Cookbook, Google Cloud, My SQL, Recipe, Subnet, Virtual Machine, Virtual Private Cloud.

I. INTRODUCTION

Chef is a configuration management technology used to automate the infrastructure provisioning. It is developed on the basis of Ruby DSL language. It is used to streamline the task of configuration and managing the company's server.
Chef does not make assumptions on the current status of a node. It uses its mechanisms to get the current status of machine.

1) Virtual Private Cloud (VPC) provides networking functionality to Compute Engine virtual machine (VM) instances, Google Kubernetes Engine (GKE) clusters, and the App Engine flexible environment. VPC provides networking for your cloud-based resources and services that is global, scalable, and flexible.
2) Google cloud platform (GCP) is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end user products for Google cloud account , it requires a bank account details
3) An instance is a virtual machine (VM) hosted on Google's infrastructure. You can create an instance by using the Google Cloud Console, the Google cloud command- line tool, or the Compute Engine API.
4) Cloud SQL for MySQL is a fully-managed database service that helps you set up, maintains, manage, and administer your MySQL relational databases on Google Cloud Platform.

II. LITERATURE SURVEY

Authors in this Juve and Deelman [1] paper say that infrastructure as a service clouds provide the ability to provision Virtual Machines on demand, but they do not give information for managing those resources which are provisioned. Hence, to use such clouds effectively, tools are needed to be used which can help users to easily deploy applications in the cloud. The authors of this paper developed a system to create, configure, and manage the Virtual Machine deployments in the cloud
Due to variety of the operating system and applications, it becomes very difficult to deploy a large number of virtual machines in a short period. This Zhang et al. [2] paper proposes an automatic deployment mechanism based on cloud computing platform open stack. This system is responsible for the automatic deployment at operating system level as well as application level. They have developed an interactive dashboard for the users which helps them to deploy their systems and the applications without professional knowledge of cloud. This Callanan et al. [3] paper has presented the architecture of an environment migration framework for automating the migration of existing infrastructure, creation, and configuration in the cloud. They have discussed some challenges faced while migrating the applications. Compute, storage, and network are the primary resources of computing. Provisioning time for deployment of these resources is remarkably minimized by virtualization technology. However, to construct a cloud-based infrastructure, still only data centre virtualization is not sufficient. If we go for only this data centre virtualization technique, then it may generate virtual resource sprawl. In addition, the infrastructure which is cloud-based cannot construct by only virtual infrastructure. In other hand, physical infrastructure also needs to be automated.

Hence, to automate and manage cloud-based infrastructure (virtual as well as physical resources), we need software. For management and automation of cloud-based infrastructure, different modules and their integration are discussed in cloud management and automation paper [4].
In this paper - Hintsch, Johannes; Görling, Carsten; and Turowski, Klaus; "A review of the literature on configuration management tools", [5] the existing research on this topic is reviewed comprehensively. Readers are provided with a descriptive analysis of the published literature as well as with an analysis of the content of the respective research works. The paper serves as an overview for researchers who are new to the topic.
Furthermore, it serves to identify work related to an intended research field and identifies research gaps. Practitioners are provided with a means to identify solutions to their organizational problems.
Author Waldemar Hummer,[6] in this paper explained that-deploying middleware in production environments, operations teams typically rely on automation logic (scripts).
Poorly written automations incur an increased risk of compromising the stability of deployments. According to author, infrastructure as code (IAC) is becoming a key concept to facilitate the development of automation logic for deploying, configuring, and upgrading inter-related middleware components. IAC automations are designed to be repeatable, making the system converge to a desired state starting from arbitrary states.
The notion of impotence builds the foundation for repeatable, robust automations. State-of-the-art IAC tools, such as chef or puppet, provide developers with abstractions to express automation steps as idempotent units of work. He demonstrated his framework for comprehensive testing of automation scripts. The demo is based on and complements this paper presents at the middleware'13 main conference. While the approach is general, the demo is tailored to chef. Throughout the demo, we showcase our tool based on testing scenarios with real-world chef scripts.
James o. Benson, john j. Prevost, and Paul rad [7] worked and explained that 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 increasingly powerful, anchoring the ecosystem of cloud services.
Cloud infrastructure services are appealing in part because they enable customers to acquire and release infrastructure resources on demand for applications in response to load surges. This paper addresses the challenge of building an effective multi-cloud application deployment controller as a customer add-on outside of the cloud utility service itself. Such external controllers must function within the constraints of the cloud providers' Apis.
James o. Benson, John j. Prevost[8] worked on a paper and came to a conclusion. In this paper, they described the different steps necessary to deploy applications using such external controller.
Then with a set of candidates for such external controllers, they used the proposed taxonomy to survey several management tools such as Chef, Saltstack, and Ansible for application automation on cloud computing services based on the defined model. Chef typically depends on a master/slave configuration where there is a main master node or server that uploads recipes and then is deployed to the clients.
Dmitry Duplyakin and Robert Ricci[9] researched , worked and explained that chef uses SSH to communicate and authenticates via the use of certificates. Companies like ancestry.com, bonobos, Etsy and Bloomberg use chef to manage their systems. There are nearly 2200 cookbooks and over 62,500 chefs supporting the supermarket on supermarket.chef.io.
In addition, chef's website interface also allows the administrator to view and search node activity, assign cookbooks, roles and nodes to tasks.
To control the clients in Chef, you can import a cookbook or create one of your own. The Cookbooks are written in Ruby, so some knowledge of Ruby Programming is very helpful, especially since official documentation can be vague at times for more advanced functions

III. EXISTING SYSTEM

Designing and developing applications means satisfying any number of separate requirements. Development decisions that are made to satisfy a requirement may affect other requirements, often in ways that are difficult to understand or predict. The failure to meet these requirements could mean the failure of a project.
Some of the problems that existed are: Manual Deployments, Complex Deployments, Inconsistent Environments, Waterfall Culture.

Manual Deployments include production deployments that occurred every three months and could take upto 8 hours to complete. To fix this, help of 5 to 10 different workers are required. It was a huge organizational challenge, and every deployment had its own unique issues.

Complex Deployments include reliability and human interaction. Due to incorrect shutdown sequence, some applications would not start. Some applications were also not configured the same on every environment, further compromising reliability. Deployments often require human interactions to perform an action on the website to deploy properly.

Our environments were simply inconsistent. This results in errors occurring on Product that did not occur on previous environments. It is painful and costly issue when your production environment is down for longer than you had expected. This was basically the "human factor".

Before Automation, the current culture was basically using waterfall techniques. Chef drives the automation of our deployments and reduces the manual complexities and inconsistencies.

IV. PROPOSED SYSTEM

1) In order to overcome the existing problems, we propose the use of automation tool "Chef" in Google Cloud.
2) Chef Automation tool, in Google Cloud, automates all the process and reduce lot of manpower along with time.
3) Chef is a configuration management program that maintains infrastructure by writing code rather than doing human tasks; allowing it to be easily automated, tested, and deployed.
4) Chef automates application setup, deployment, and administration throughout the network, regardless of whether it is cloud or hybrid. Additionally, Chef may be used to accelerate application deployment.
5) A chef is an excellent tool for speeding software delivery; software development speed refers to how rapidly the program may change in response to new needs or situations. So, we use Chef Automation in Google Cloud to overcome the delayed speed.
6) Accelerating software delivery, once your infrastructure is automated, all software needs like testing, developing new environments for software deployments and so on becomes more fast.
7) Enhanced service Resilience, in Google Cloud, is achieved by automating the infrastructure, which monitors for defects and failures before they exist and also allows for faster recovery from errors.
8) In Google Cloud, Chef reduces risk and enhances compliance at all stages of deployment. It lowers disputes in the development and manufacturing environments.
9) Y67dChef can readily be converted to a cloud environment and servers. Infrastructure can be quickly designed, installed, and managed automatically by Chef.

A. Algorithm

1) In Google cloud platform enable the compute engine, SQL and IAM roles services.
2) Create a chef workstation on Google cloud. Use the workstation to create a compute engine, VM cloud, SQL instance, firewall rules and a database with chef. Before we begin, select or create a Google Cloud Project.
3) Go to the project page and enable billing for your project.
4) Create a new service account. Add compute engine and cloud SQL permission to the account. Use Google Cloud to create a new compute engine instances with service account you just created and add SSH to it.
5) From the chef workstation machine, install the chef workstation and GIT. Create cookbook folder.
6) Chef requires a GIT repository with at least one commit, so set up GIT and make an initial commit to cookbook directory.
7) Generate the chef book to specify your infrastructure. Add the dependency on the Google Cloud cookbook to the metadata.rb file of the cookbook.
8) Install the Google Cloud cookbook from the supermarket using the knife tool. Create a copy of your chef workstation service account key on the chef workstation machine.
9) This will be used by the chef cookbooks you created. Use the chef Google cloud cookbooks to define your compute engine infrastructure.
10) Create a recipe within the cookbook, including the compute engine recipe to your infrastructure cookbooks by using default recipe.
11) Run the chef client to run the cookbook. Finally, we can see the creation of VM, SQL and VPC in the GCP console.

B. Architecture

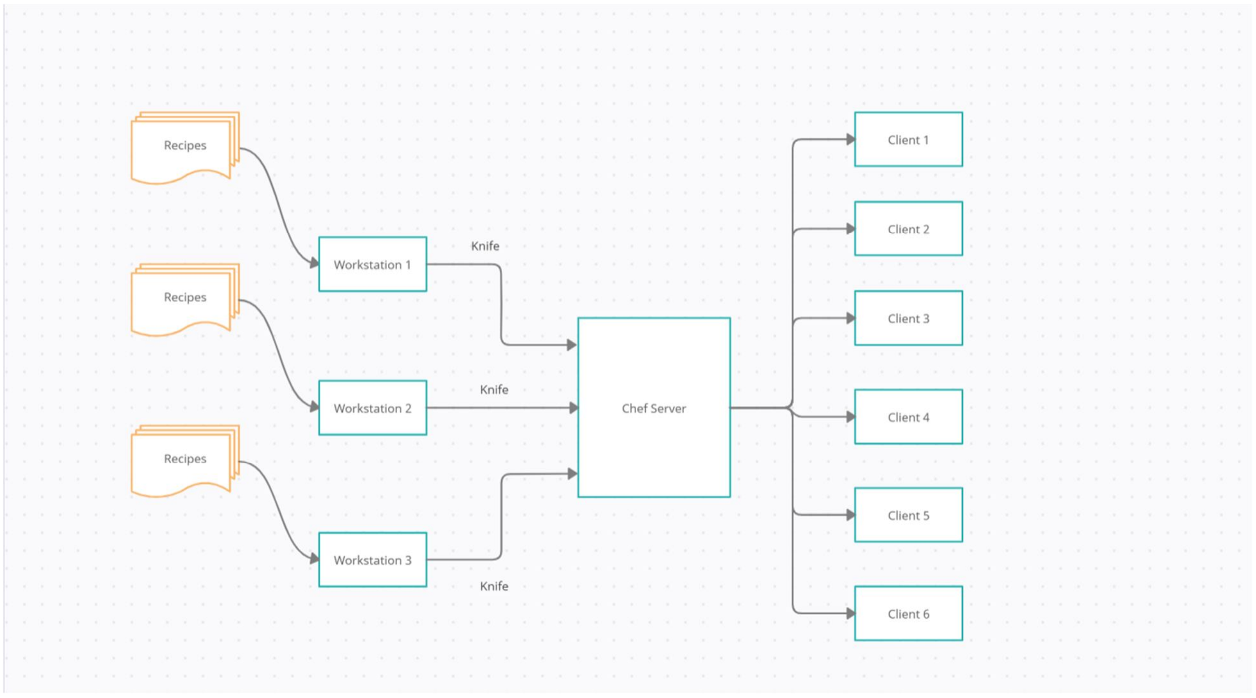


Fig. 1 Architecture

C. Project Description

Chef Infra is a powerful automation platform that transforms infrastructure into code. Whether you're operating in the cloud, on-premises, or in a hybrid environment, Chef Infra automates how infrastructure is configured, deployed, and managed across your network, no matter its size.

This diagram shows how you develop, test, and deploy your Chef Infra code.

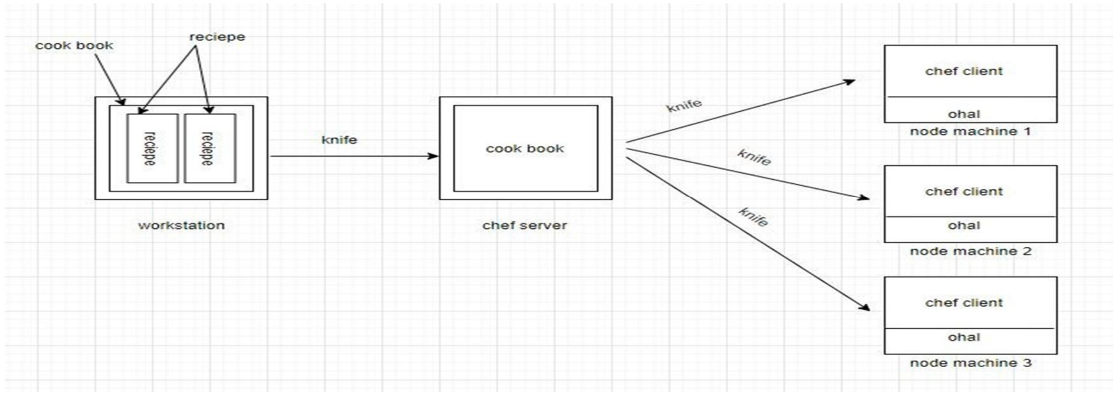


Fig. 2 Diagram of developing, testing, and deploying Chef Infra code

1) Chef Workstation is the location where users interact with Chef Infra. With Chef Workstation, users can author and test cookbooks using tools such as Test Kitchen and interact with the Chef Infra Server using the knife and chef command line tools.
2) Chef Infra Client nodes are the machines that are managed by Chef Infra. The Chef Infra Client is installed on each node and is used to configure the node to its desired state.
3) Chef Infra Server acts as a hub for configuration data. Chef Infra Server stores cookbooks, the policies that are applied to nodes, and metadata that describes each registered node that is being managed by Chef. Nodes use the Chef Infra Client to ask the Chef Infra Server for configuration details, such as recipes, templates, and file distributions.

D. Implementation Steps

1) *Step 1:* Create a Chef Workstation on a Compute Engine VM

a) Specify default settings for the Cloud SDK.

```
gcloud services enable compute.googleapis.com
gcloud services enable sqladmin.googleapis.com
gcloud services enable iam.googleapis.com
gcloudconfigset compute/region us-west1
gcloudconfigset compute/zone us-west1-a
```

b) Create the service accounts that Chef will use:

1. Create Compute Engine service account
    a) Create a new service account

```
gcloudiam service-accounts create chef-workstation
```

    b) Add Compute Engine and Cloud SQL permissions to the account

2. Use gcloud to create a new Compute Engine instance with the service account just created and SSH it.

```
gcloud compute instances create chef-workstation\
--image-familydebian-9—image-project debian-cloud\
--service-account=chef-workstation@$PROJECT_NAME.iam.gserviceaccount.com\
--scopes=https://www.googleapis.com/auth/cloud-platform
gcloud compute ssh chef-workstation
```

3. From the chef-workstation machine, install the Chef Workstation and git

```
echo "PROJECT_NAME=$PROJECT_NAME">> ~/.bashrc
wget https://packages.chef.io/files/stable/chef-workstation/0.1.162/ubuntu/18.04/chef-
workstation_0.1.162-1_amd64.deb
sudodpkg –I chef-workstation_0.1.162-1_amd64.deb
sudo apt update
sudo apt install git –y
```

2) *Step 2:* Create a Chef Workstation on a Compute Engine VM

a) Specify default settings for the Cloud SDK.

```
gcloud services enable compute.googleapis.com
gcloud services enable sqladmin.googleapis.com
gcloud services enable iam.googleapis.com
gcloudconfigset compute/region us-west1
gcloudconfigset compute/zone us-west1-a
```

b) Create the service accounts that Chef will use:

1. Create Compute Engine service account
    a) Create a new service account

```
gcloudiam service-accounts create chef-workstation
```

    b) Add Compute Engine and Cloud SQL permissions to the account

2. Use gcloud to create a new Compute Engine instance with the service account just created and SSH it.

```
gcloud compute instances create chef-workstation\
--image-familydebian-9—image-project debian-cloud\
--service-account=chef-workstation@$PROJECT_NAME.iam.gserviceaccount.com\
--scopes=https://www.googleapis.com/auth/cloud-platform
gcloud compute ssh chef-workstation
```

3. From the chef-workstation machine, install the Chef Workstation and git

```
echo "PROJECT_NAME=$PROJECT_NAME">> ~/.bashrc
wget https://packages.chef.io/files/stable/chef-workstation/0.1.162/ubuntu/18.04/chef-workstation_0.1.162-1_amd64.deb
sudodpkg –I chef-workstation_0.1.162-1_amd64.deb
sudo apt update
sudo apt install git –y
```

c) Generate the Chef cookbook to specify your infrastructure.

```
chef generate cookbook infrastructure
```

d) Add the dependency on the google-cloud cookbook to the metadata.rb file of the cookbook

```
echo ”depends ‘google-cloud’,’~>0.4.0’”>>infrastructure/metadata.rb
```

e) Install Google Cloud cookbooks from the supermarket using the knife tool

```
knife cookbook site install google-cloud –o
```

f) Create a copy of your chef-workstation service account key on the chef workstation machine. This will be used by chef cookbooks created

```
gcloudiam service-accounts keys create ~/key.json-iam-account=chef-workstation@$PROJECT_NAME.iam.gserviceaccount.com
```

3) Step 3: Create Compute Engine Cookbook

(i) Create a new recipe within the infrastructure cookbook.

```
chef generate recipe infrastructure compute
```

(ii) Write the code for creating VM instance and VPC network in GCP console.

(iii) Include the Compute Engine recipe to your infrastructure cookbook's default recipe**.**

```
echo"include_recipe'infrastructure::compute'">>infrastructure/recipes/default.rb
```

4) Step 4: Create cloud SQL Cookbook

(i) Create a new recipe within the infrastructure cookbook

```
chef generate recipe infrastructure sql
```

(ii) Write the code for creating MySQL in GCP console.

(iii) Include the Cloud SQL recipe into your infrastructure cookbook's default recipe.

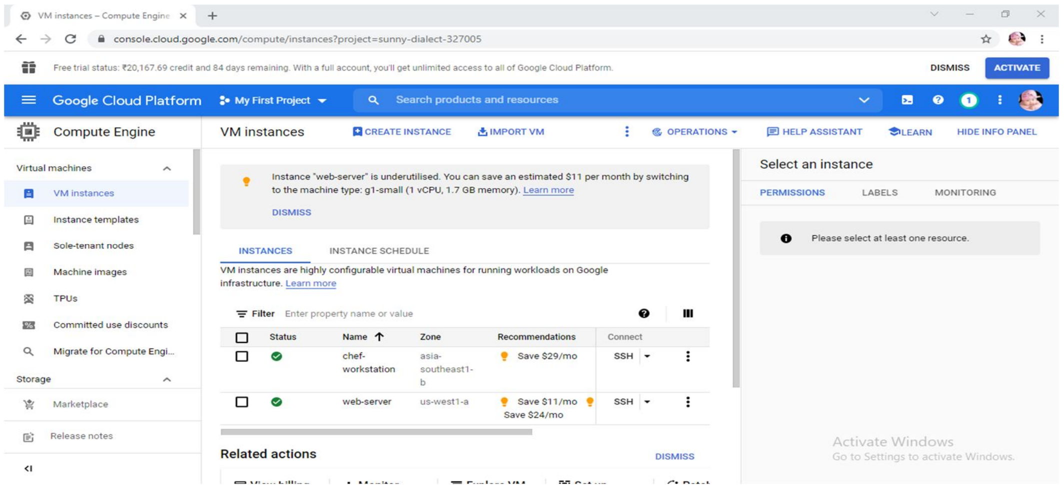
```
echo ”include recipe ’infrastructure::sql’”>>infrastructure/recipes/default.rb
```

5) Step 5: Run the infrastructure cookbook

(i) Run chef-client in local mode with the infrastructure cookbook as the run-list.

```
chef-client –z –o infrastructure
```

V. RESULT ANALYSIS



Free trial status: ₹20,167.69 credit and 84 days remaining. With a full account, you'll get unlimited access to all of Google Cloud Platform.

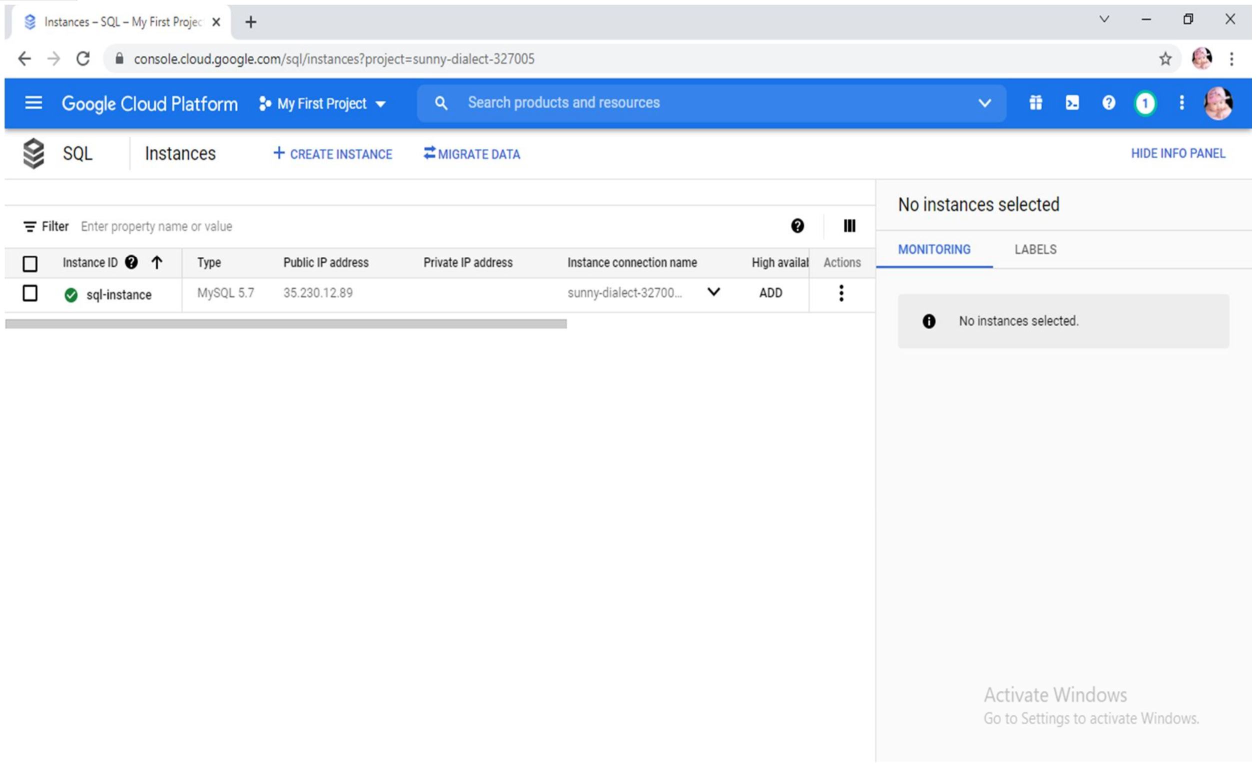
Instance "web-server" is underutilised. You can save an estimated $11 per month by switching to the machine type: g1-small (1 vCPU, 1.7 GB memory). Learn more

VM instances are highly configurable virtual machines for running workloads on Google infrastructure. Learn more

Status	Name	Zone	Recommendations	Connect
✓	chef-workstation	asia-southeast1-b	Save $29/mo	SSH
✓	web-server	us-west1-a	Save $11/mo Save $24/mo	SSH

Please select at least one resource.

Fig. 3 Output of Chef-Workstation and Web-Server running with active status

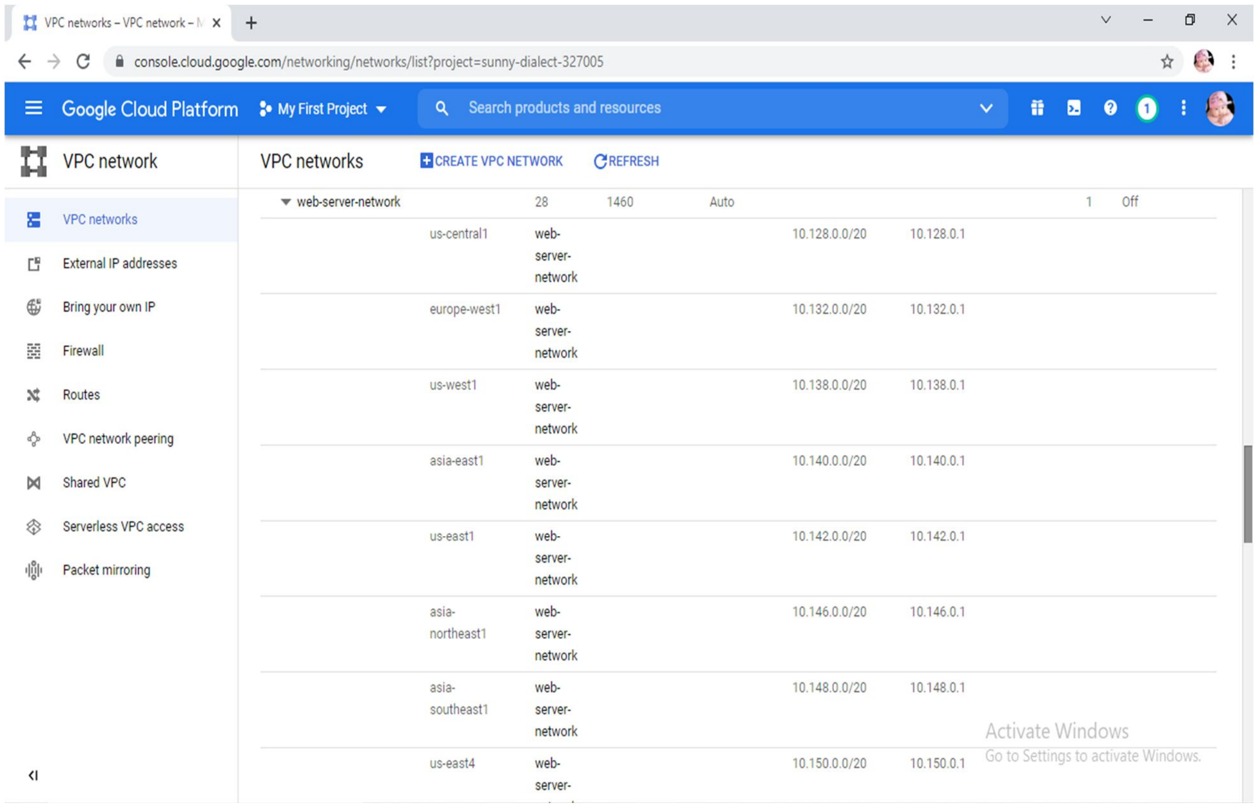

Instances – SQL – My First Projec
console.cloud.google.com/sql/instances?project=sunny-dialect-327005
Google Cloud Platform
My First Project
Search products and resources
SQL
Instances
CREATE INSTANCE
MIGRATE DATA
HIDE INFO PANEL
Filter
Enter property name or value
MONITORING
LABELS
Activate Windows
Go to Settings to activate Windows.

Instance ID	Type	Public IP address	Private IP address	Instance connection name	High availal	Actions
sql-instance	MySQL 5.7	35.230.12.89		sunny-dialect-32700...	ADD	

No instances selected

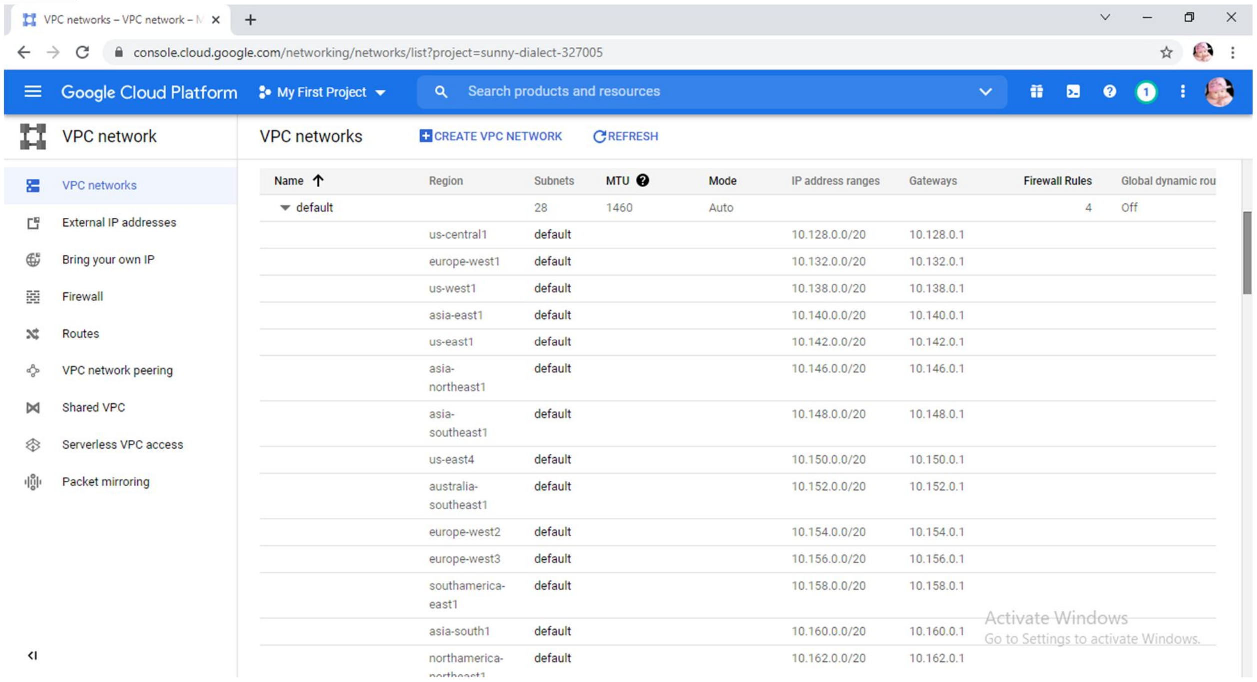
No instances selected.

Fig. 4 Output of SQL instance running with active status



web-server-network		28	1460	Auto		1	Off
	us-central1	web-server-network		10.128.0.0/20	10.128.0.1		
	europe-west1	web-server-network		10.132.0.0/20	10.132.0.1		
	us-west1	web-server-network		10.138.0.0/20	10.138.0.1		
	asia-east1	web-server-network		10.140.0.0/20	10.140.0.1		
	us-east1	web-server-network		10.142.0.0/20	10.142.0.1		
	asia-northeast1	web-server-network		10.146.0.0/20	10.146.0.1		
	asia-southeast1	web-server-network		10.148.0.0/20	10.148.0.1		
	us-east4	web-server-		10.150.0.0/20	10.150.0.1		

Fig. 5 Output of VPC running with active status



VPC networks

Name	Region	Subnets	MTU	Mode	IP address ranges	Gateways	Firewall Rules	Global dynamic rou
default		28	1460	Auto			4	Off
	us-central1	default			10.128.0.0/20	10.128.0.1		
	europe-west1	default			10.132.0.0/20	10.132.0.1		
	us-west1	default			10.138.0.0/20	10.138.0.1		
	asia-east1	default			10.140.0.0/20	10.140.0.1		
	us-east1	default			10.142.0.0/20	10.142.0.1		
	asia-northeast1	default			10.146.0.0/20	10.146.0.1		
	asia-southeast1	default			10.148.0.0/20	10.148.0.1		
	us-east4	default			10.150.0.0/20	10.150.0.1		
	australia-southeast1	default			10.152.0.0/20	10.152.0.1		
	europe-west2	default			10.154.0.0/20	10.154.0.1		
	europe-west3	default			10.156.0.0/20	10.156.0.1		
	southamerica-east1	default			10.158.0.0/20	10.158.0.1		
	asia-south1	default			10.160.0.0/20	10.160.0.1		
	northamerica-northeast1	default			10.162.0.0/20	10.162.0.1		

Fig. 6 Output of Subnet running with active status

Virtual Machine, MySQL, Virtual Private Cloud and Subnet has been successfully created and installed by using **Chef Automation tool.**

VI. CONCLUSIONS

A. The manual installation, updating and configuration of software packages require 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.

B. This study of chef automation tool defines the importance to achieve architecture for the distributed systems. The future work includes the deploying and managing the infrastructure and applications; on top of that using, this chef automation tool analyze the security, repeatability, reliability, and scalability impacts on the deployed distributed system.

REFERENCES

[1] NicolÃ¡s Paez, Versioning Strategy for DevOps Implementations, Department of Science and Technology Universidad Nacional de Tres de Febrero, IEEE 2020, pp. 1-6.

[2] Morgan B. Kamuto, Josef J. Langerman, Factors Inhibiting the Adoption of DevOps in Large Organisations: South African Context, 2020 2nd IEEE International Conference On Recent Trends In Electronics Information & Communication Technology, pp. 48-51

[3] Dmitry Duplyakin and Robert Ricci, Introducing Configuration Management Capabilities into CloudLab Experiments, IEEE INFOCOM International Workshop on Computer and Networking Experimental Research Using Testbeds, 2020, pp. 453-458

[4] 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, 2019, pp. 295-302.

[5] James O. Benson, John J. Prevost, and Paul Rad, Survey of Automated Software Deployment for Computational and Engineering Research, IEEE Transactions, 2019.

[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, 2019, 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, 2020, pp. 295-302.

[8] Nishant Kumar Singh, Sanjeev Thakur, Himanshu Chaurasiya and Himanshu Nagdev, Automated Provisioning of Application in IAAS Cloud using Ansible Conguration Management,1st International Conference on Next Generation Computing Technologies (NGCT-2015) Dehradun, India, 4-5 September 2018, pp. 81-85.

[9] Johannes Hintsch, Carsten GÂ¨orling, and Klaus Turowski, Modularization of Software as a Service Products: A Case Study of the Conguration Management Tool Puppet, Third International Conference on Enterprise Systems, 2019, pp. 184-191