Course Code: INFO3606
Course Title: Cloud Computing
Semester Project

Tevin Achong - 816000026, Jonathan Owen -

April 8, 2020

Contents

1 Chef **2**
- 1.1 Components . . . 2
- 1.2 Architecture . . . 3
- 1.3 Features . . . 3
- 1.4 Official Website . . . 4
- 1.5 Latest Version . . . 4
- 1.6 Advantages . . . 4
- 1.7 Disadvantages . . . 5

2 Ansible **6**
- 2.1 Components . . . 6
- 2.2 Architecture . . . 6
- 2.3 Features . . . 6
- 2.4 Official Website . . . 6
- 2.5 Latest Version . . . 6
- 2.6 Advantages . . . 6
- 2.7 Disadvantages . . . 6

3 Fuel **7**
- 3.1 Components . . . 7
- 3.2 Architecture . . . 7
- 3.3 Features . . . 7
- 3.4 Official Website . . . 7
- 3.5 Latest Version . . . 7
- 3.6 Advantages . . . 7
- 3.7 Disadvantages . . . 7

4 Puppet **8**
- 4.1 Components . . . 8
- 4.2 Architecture . . . 8
- 4.3 Features . . . 8
- 4.4 Official Website . . . 8
- 4.5 Latest Version . . . 8
- 4.6 Advantages . . . 8
- 4.7 Disadvantages . . . 8

5 Compass **9**
- 5.1 Components . . . 9
- 5.2 Architecture . . . 9
- 5.3 Features . . . 9
- 5.4 Official Website . . . 9
- 5.5 Latest Version . . . 9
- 5.6 Advantages . . . 9
- 5.7 Disadvantages . . . 9

6 References NEED TO FORMAT PROPERLY **10**

1 Chef

Chef is a configuration management tool that is used to streamline the task of configuring and maintaining a company's servers. Aditionally, Chef can integrate with various cloud-based platforms to automatically provision and configure new machines. It contains solutions for small and large scale systems, and features and pricing for each of the various ranges. Essentially, Chef ensures that the files and the software that users are expecting to be on a machine are actually present, configured correctly, and working as is expected. Performing these tasks for a single machine is fairly straightforward. However, as an organization's infrastructure scales up (more machines are introduced) it becomes increasingly difficult. This is the reason Chef was developed and is used.

1.1 Components

Chef operates with three core components: **Chef Server**, **Workstations**, and **Nodes**. These three components communicate in a linear fashion, as explained below.

1. **Workstations:** All the configuration code is created, tested, and changed at workstations, which are personal computers or virtual servers. There can exist as many workstations as is necessary. Additionally, cookbooks and policies that will be pushed to the Chef server and pulled by nodes are tested and maintained here. The Chef Workstation provides chef and command line tools, the testing tools Test Kitchen, ChefSpec, Cookstyle, and Foodcritic, and InSpec - a tool that allows you to write automated tests for compliance, security and policy requirements. Cookbooks created on workstations can be used privately or by one organization, or uploaded to the Chef Supermarket for other to use. Workstations can also be used to download cookbooks created by other Chef users and found in the Chef Supermarket.

2. **Nodes:** These are the servers that are managed by Chef i.e. the machines to which changes are being pushed. Nodes are generally a fleet of multiple machines that require the benefits of automation. Chef can manage nodes that are virtual servers, containers, network devices, and storage devices and a Chef client is installed on every node that is under management by Chef.

3. **Chef Server:** The Chef Server is the center of all of Chef's operations. It provides a communication pathway between the workstations (where infrastructure is coded) and the nodes (where the configurations are deployed by the Chef client). Any configuration files, metadata, cookbooks and other information created on workstations are stored on the Chef server. Aditionally, the Chef server contains information regarding the state of all nodes at the time of the last chef-client run. Any changes made to infrastructure code must pass through the Chef server in order to be applied to nodes. Before accepting changes, the Chef server authenticates all communication via its REST API using public key encryption. In turn, the Chef Server is also made of several components which aid it in efficiently communicating with workstations and nodes. Each Chef Server uses an NGINX front-end load balancer to route all requests to the Chef Server API, and PostgreSQL to store data. A web interface, known as Chef manage, is used for common Chef server management tasks. An Apache Solr instance, wrapped by chef-solr, is used for indexing and searching. All these components help to make the Chef server capable of

handling requests for several thousands of nodes and make Chef server a resource heavy application.

1.2 Architecture

In order to achieve its goals, Chef treats infrastructure as code. Instead of manually changing anything, the machine setup is described in a Chef *recipe*. *Cookbooks* store collections of recipes. Ideally, one cookbook relates to a single task, but it can have numerous server configurations involved. The chef server stores each of the cookbooks and as a new chef client node checks in with the server, recipes are sent to tell the node how to configure itself. Afterwards, the client will occassionally check in with the server to see if anything needs to be changed. If something does need to be changed, then the client deals with it. Patches and updates can be applied to the entire infrastructure by changing the recipe; there is no need to interact with each machine individually. *Bookshelf* is used to store cookbooks and related files and templates. The Chef Server uses a Bookshelf that operates as a versioned repository. Full root access is required. Whenever a cookbook is uploaded to the Chef server, the new version of the cookbook is compared to the one already stored on the server. If any changes exist, a new version is stored. If resources are shared between cookbooks and cookbook versions, they will not be stored more than once. This is because the Chef Server stores one copy of a file or template. As such, Chef utilizes a three-tier client-server architecture. The working units (like cookbooks) are developed on the Chef workstation. From the command line utilities like Knife, they are uploaded to the Chef server and all the nodes which are present in the architecture are registered with the Chef server. Figure 1 below depicts this architecture.

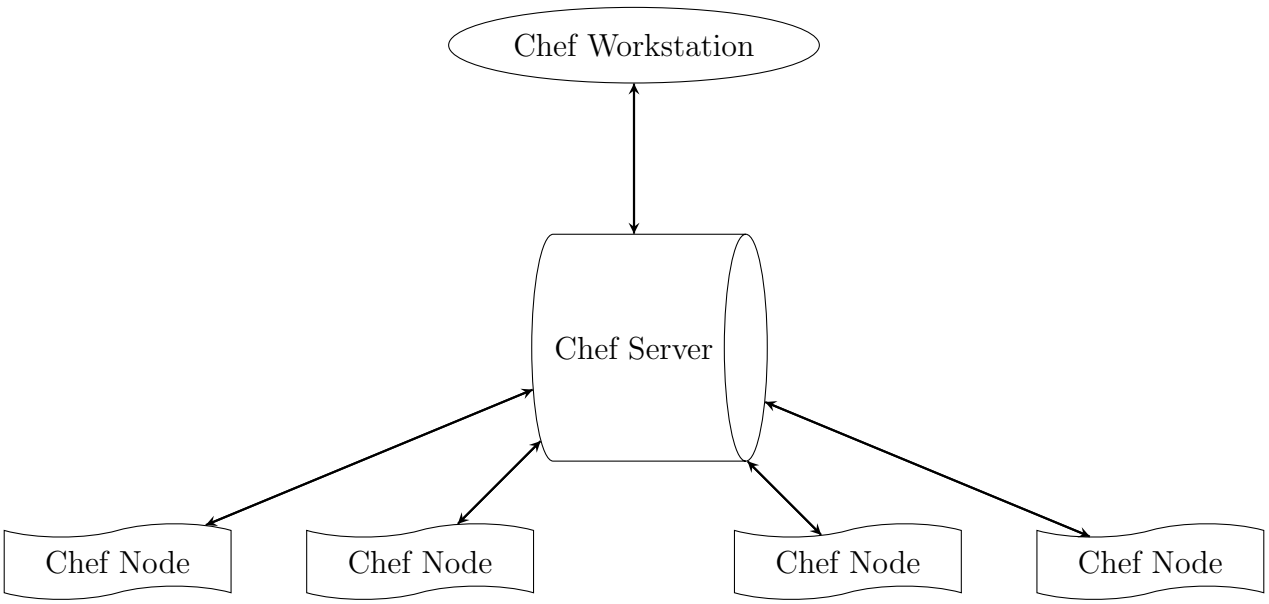

Figure 1: Architecture of Chef

1.3 Features

Chef provides the following features:

- Automatic Backup
- Automatic Notifications
- Compliance Management
- Configuration Management
- Data Recovery
- Data Visualization
- Real Time Analytics
- Real Time Data
- Real Time Monitoring
- Real Time Notifications
- Real Time Reporting
- Real Time Updates
- Server Monitoring

1.4 Official Website

The official website for Chef is located at https://www.chef.io.

1.5 Latest Version

As of January 28, 2019, the latest stable release of the Chef client is v14.10.9.
As of October 15, 2018, the latest stable release of the Chef server is v12.18.14.

1.6 Advantages

Advantages of using Chef include:

1. There is a low barrier for entry. Chef uses native Ruby language for configuration. This means that it can be easily picked up by anyone who has some development experience.
2. It integrates execellently with cloud. Using the knife utility, Chef can be easily integrated with most existing cloud technologies. It is a suitable tools for organizations that wish to distribute their infrastructure on multi-cloud environment.
3. Chef can help organizations to accelerate software delivery, i.e. how quickly the software is able to be changed in response to new requirements or conditions.
4. Helps organizations to catch bugs and issues before they occur. Infrastructure automation increases a system's resiliency just as much as it accelerates delivery speed.
5. Improving risk management - the *quality* of changes made to the software.
6. When using Chef, you can deliver all your infrastructure everywhere continuously.

1.7 Disadvantages

Some disadvantages include:

1. Cookbooks require constant monitoring so that the people who are working do not cause any issues with them.
2. Only Chef solo is available.
3. Currently, Chef is only a good fit for Amazon Web Services Cloud.
4. Someone who is not familiar with Ruby may have a hard time learning it.
5. There is currently a lack of proper documentation.

2 Ansible

2.1 Components

2.2 Architecture

2.3 Features

2.4 Official Website

2.5 Latest Version

2.6 Advantages

2.7 Disadvantages

3 Fuel

3.1 Components

3.2 Architecture

3.3 Features

3.4 Official Website

3.5 Latest Version

3.6 Advantages

3.7 Disadvantages

4 Puppet

4.1 Components

4.2 Architecture

4.3 Features

4.4 Official Website

4.5 Latest Version

4.6 Advantages

4.7 Disadvantages

5 Compass

5.1 Components

5.2 Architecture

5.3 Features

5.4 Official Website

5.5 Latest Version

5.6 Advantages

5.7 Disadvantages

6 References NEED TO FORMAT PROPERLY

https://shadow-soft.com/chef-benefits/
https://www.tutorialspoint.com/chef/chef_overview.htm
https://www.linode.com/docs/applications/configuration-management/beginners-guide-chef/