**CHEF**

**Configuration Management Tool**

**CHEF** Ruby or Erlang do language se milkar bna h.

Chef is **an automation tool**. It is a Configuration Management (CM) tool that lets you automate processes and tasks across numerous servers and other devices of an organization in simple steps. Such a framework is highly beneficial to a company. **(OPERATION TEAMS PART)**

**TWO TYPES OF CONFIGURATION MANAGEMENT TOOLS.**

1. **PUSH BASED** – Push config serve pushes config. to the nodes. **(Ansible, Salt Stack)**
2. **PULL BASED** – Pull config. nodes check with the server periodically and fetches the config. From it. **(Chef, Pupet)**

**IAC (INFRASTRUCTURE AS CODE)**

**A picture containing graphical user interface

Description automatically generated**

knife

knife

knife

knife

**Workstation**

* Workstation are personal computers or virtual servers where all config. Code is created, tested or changed.
* Devops engineer actually sits here and write codes this code is called recipe. A collection of recipes are known as cookbook.
* Workstation communicate with the chef server using knife.
* Knife is a command line took that uploads the cookbook to the server.

**Chef-Server**

* the chef-server is a middle man between workstation and the nodes.
* All cookbooks are stored here.
* Server may be hosted locally or remote.

**Node**

* Nodes are the systems that require the config.
* Ohai fetches the current state of the node its located in.
* Node communicate with the chef-server using the chef-client.
* Each node can have a different config. Required.
* Chef-client is installed on every node.

**Chef workstation –** where you write code.

**Chef server –** where you upload code.

**Chef node -**  where you apply code.

**Knife –** tool to establish communication among workstation, server and node knife is a command line tool that runs on workstation.

**Chef client –** tool runs on every chef node to pull code from chef server.

**Chef client will**

* gather current system config.
* Download the desired system config. From the chef server.
* Configure the node such that it adhere to the policy.

**Ohai –** maintain current state information of chef code.

**Idempotency –** tracking the state of system resources to ensure that the changes should not reapply repeatedly.

**Chef supermarket –** where you get custom code.

**Recipe =** Infrastructure code. Recipe cookbook me hoti h. recipe me code likha hota h.

**Recipe** ka code Ruby me likha hota h.

**Cookbook =** collection of recipe. Cookbook workstation pe hoti h lekin save chef server pe hoti h.

**Cookbook stores on server.**

**Knife –** knife code ko workstation se chef server pe push krne ke lie use krte h or chef server se node pe push krne ke lie krte h.

**Bootstrap –** Server jab node se connect hota h to us process ko bootstrap khte h.

**Chef Supermarket –** yhan aapko sabhi tarike ki recipe or cookbook mil jayegi.

**Chef** workstation or node dono jagha install hoga.

**Node** pe Ohai or Chef-client do chije hoti h.

**Ohai** node ka database hota h.

**Idempotency** ohai new file or new code ko hi pickup krega old data ko nhi jo uske pas already rakha hua h. mtlb hum bar-bar kisi code ko repeat nhi krte.