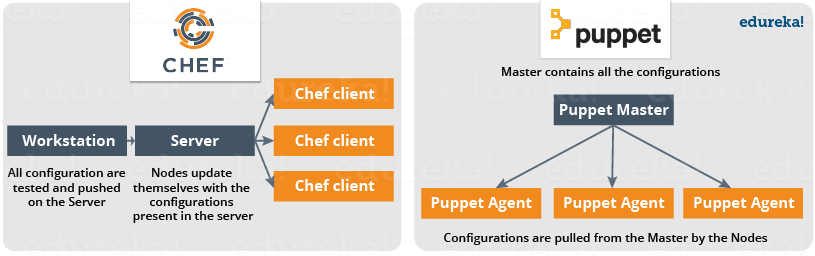
CHEF

**Introduction:**

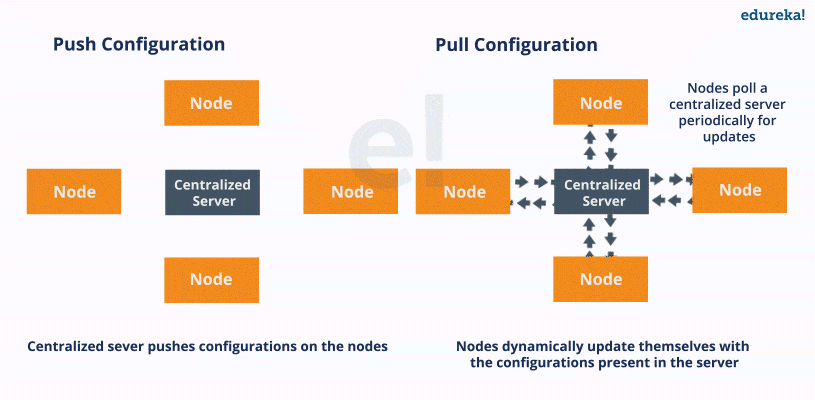
* Chef is a tool used for Configuration Management and is closely competing with Puppet.
* Chef is an automation tool that provides a way to define infrastructure as code.
* Chef uses a pure-Ruby, domain-specific language (DSL) for writing system configurations.
* Automations done by Chef
  + Infrastructure Configuration (Configurations across all networks)
  + Application Deployment
* Chef is Client-Server model but has WorkStation



* In this tool, Nodes will pull configurations from the Central Server dynamically in regular intervals.

**Configuration Management:**

* As the name states it is Managing of the configurations for the systems/server.
* Tools like Chef and Puppet helps to automate the configuration process for ‘N’ servers in the environment.
* It reduces the task and time for configuring systems. Once the configuration are punched in the central server all the child nodes pulls the needed information.
* It allows access to an accurate historical record of system state for project management and audit purposes.
* Configuration Management helps in performing the below tasks in a very structured and easy way:
  + Figuring out which components to change when requirements change.
  + Redoing an implementation because the requirements have changed since the last implementation.
  + Reverting to a previous version of the component if you have replaced with a new but flawed version.
  + Replacing the wrong component because you couldn’t accurately determine which component was supposed to be replaced.
* We have two ways to manage configurations:
  + Pull Configuration: In this type, nodes poll a central server and pull configurations from the centralized server. Ex: Chef, Puppet, etc.,
  + Push Configuration: In this type, centralized server pushes the configurations to all the nodes attached. Ex: Ansible

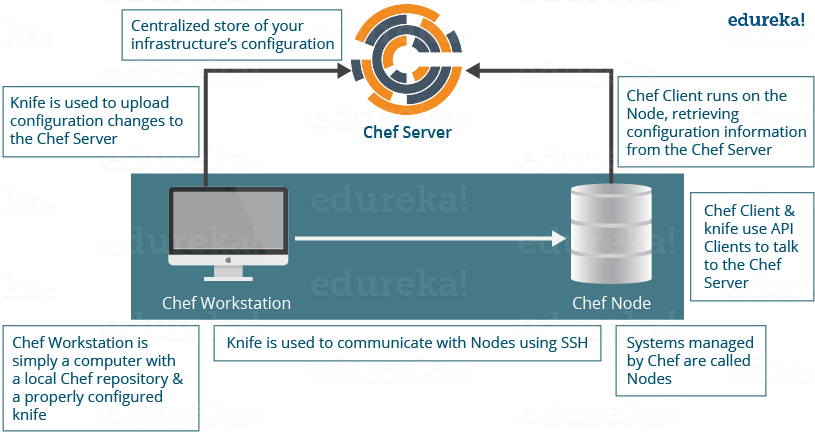


**What is Chef:**

* Chef is configuration management tool.
* Chef allows you to dynamically provision and de-provision your infrastructure on demand to keep up with the peaks in usage and downtime.
* Chef enables new services and features to be deployed and updated more frequently with less risk of downtime.
* With chef we can take advantage of the cloud features also.
* Chef enable the provision to use Infrastructure as Code.

**Chef Architecture:**

* We have 3 major Chef components
  + WorkStation
  + Server
  + Nodes



* Workstation:
  + It is where all the Chef configurations are managed.
  + These machines hold configurations which would be further pushed to the Chef server.
  + Knife is the command line utility present in WorkStation using which we can connect to Chef Server.
  + Configurations can be pushed to central Chef Server from multiple work stations.
  + D
  + Functionalities that a Work Station implements:
    - *Writing cookbooks and recipes which would be pushed to central Chef server.*
      * *Recipes*
        + It can be said as collection of resources that describe particular configuration or policy.
        + This describes how Chef manages applications and utilities and how they are to be configured.
        + These Recipes describe a series of resources that should be in a particular state, i.e. Packages that should be installed, services that should be running, or files that should be written.
      * *Cookbooks*
        + Multiple Recipes grouped together to form a cookbook.
        + A cookbook defines a scenario and contains all those required to support scenario:

Recipes, which specifies the resources to use and the order in which they are to be applied

Attribute values

File distributions

Templates

Extensions to Chef, such as libraries, definitions, and custom resources

* + - *Managing nodes on central Chef Server.*
  + We have two major components in Work Stations:
    - *Knife Utility*
      * Command line tool to communicate with central Chef Server from workstation.
      * Adding, Removing, changing configuration of nodes can be done using this utility.
      * We can update cookbooks and manage roles, environments.
    - *Local Chef Repository*
      * Place where every configuration component of the central Chef Server is present and this can be synchronized with main server repository using Knife utility.
* Chef Server
  + This acts a hub for configuration data, which contains cookbooks, and policies that are applied to nodes and metadata that describes each registered node managed by Chef-Client.
  + Chef-Client is used by nodes to get information (Recipes, Templates, and file distributions) from Chef Server.
  + Each node has chef-client installed on them which will pull the configuration from server and does the configuration on its own.
* Chef Nodes (Chef Client)
  + Nodes can be on the cloud based virtual or physical server which are managed by using Chef Server.
  + Nodes need to have agent installed on them which connects to central Chef server
  + Functions that Chef client undertakes
    - It is responsible for interacting with the central Chef Server.
    - It manages the initial registration of the Node to the central Chef Server.
    - It pulls down Cookbooks, and applies them on the Node, to configure it.
    - Periodic polling of the central Chef Server to fetch new configuration items, if any.

Advantages of Chef:

This Chef tutorial will be incomplete if, I don’t include the key benefits of Chef:

* You can automate an entire infrastructure using Chef. All tasks that were manually being done, can now be done via Chef tool.
* You can configure thousands of nodes within minutes using Chef.
* Chef automation works with the majority of the public cloud offerings like [AWS](http://www.edureka.co/blog/amazon-aws-tutorial/?utm_source=blog&utm_medium=content-link&utm_campaign=chef-tutorial).
* Chef will not only automate things, but will also keep the systems under consistent check, and confirm that the system is in fact configured the way it is required (Chef Agent/Client does this job). If somebody makes a mistake by modifying a file, Chef will correct it.
* An entire infrastructure can be recorded in the form of a Chef repository, that can be used as a blueprint to recreate the infrastructure from scratch.