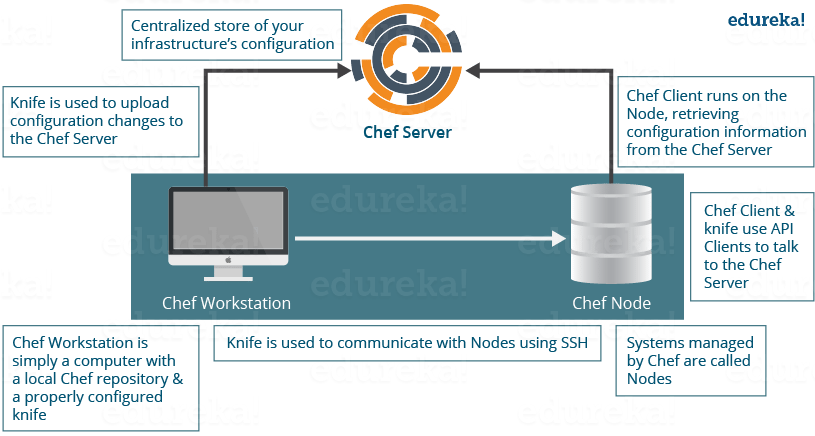
**Chef Tutorial – Chef Architecture**

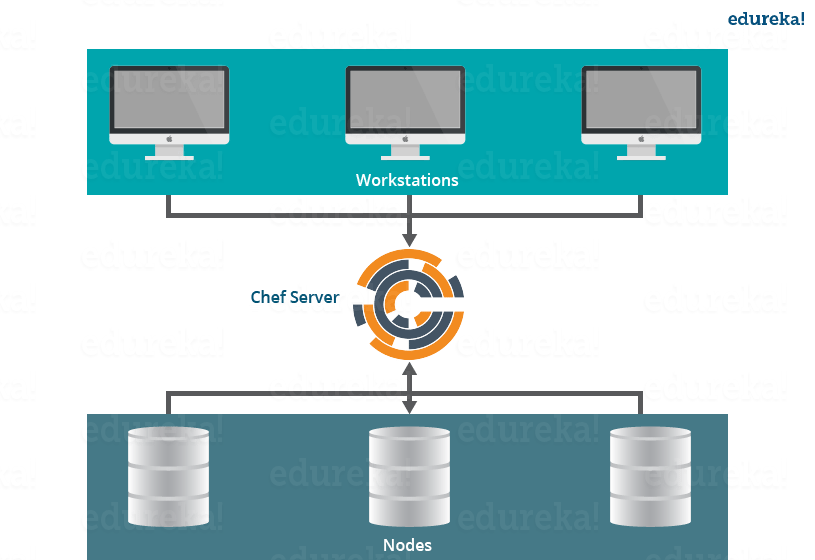
As shown in the diagram below, there are three major Chef components:

* Workstation
* Server
* Nodes



**Chef Tutorial – Workstation**

The Workstation is the location from which all of Chef configurations are managed. This machine holds all the configuration data that can later be pushed to the central Chef Server. These configurations are tested in the workstation before pushing it into the Chef Server. A workstation consists of a command-line tool called **Knife,** that is used to interact with the Chef Server. There can be multiple Workstations that together manage the central Chef Server.



Workstations are responsible for performing the below functions:

* **Writing Cookbooks and Recipes that will later be pushed to the central Chef Server**
* **Managing Nodes on the central Chef Server**

Now, let us understand the above mentioned points one by one.

**Writing Cookbooks and Recipes that will later be pushed to the central Chef Server**

**Recipes:**A Recipe is a collection of resources that describes a particular configuration or policy. It describes everything that is required to configure part of a system. The user writes Recipes that describe how Chef manages applications and utilities (such as Apache HTTP Server, MySQL, or Hadoop) 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.

*Later in the blog*, I will show you how to write a Recipe to install Apache2 package on Chef Nodes by writing a ruby code in Chef Workstation.

**Cookbooks:**Multiple Recipes can be grouped together to form a Cookbook. A Cookbook defines a scenario and contains everything that is required to support that 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 the central Chef Server**

The Workstation system will have the required command line utilities, to control and manage every aspect of the central Chef Server. Things like adding a new Node to the central Chef Server, deleting a Node from the central Chef Server, modifying Node configurations etc can all be managed from the Workstation itself.

Now let us see, what components of Workstation are required to perform the above functions.

**Workstations have two major components:**

**Knife utility:** This command line tool can be used to communicate with the central Chef Server from Workstation. Adding, removing, changing configurations of Nodes in a central Chef Server will be carried out by using this Knife utility. Using the Knife utility, Cookbooks can be uploaded to a central Chef Server and Roles, environments can also be managed. Basically, every aspect of the central Chef Server can be controlled from Workstation using Knife utility.

**A local Chef repository:** This is the place where every configuration component of central Chef Server is stored. This Chef repository can be synchronized with the central Chef Server (again using the knife utility itself).

**Chef Tutorial – Chef Server**

The Chef Server acts as a hub for configuration data. The Chef Server stores Cookbooks, the policies that are applied to Nodes, and metadata that describes each registered Node that is being managed by the Chef-Client.

Nodes use the Chef-Client to ask the Chef Server for configuration details, such as Recipes, Templates, and file distributions. The Chef-Client then does as much of the configuration work as possible on the Nodes themselves (and not on the Chef Server). Each Node has a Chef Client software installed, which will pull down the configuration from the central Chef Server that are applicable to that Node. This scalable approach distributes the configuration effort throughout the organization.

**Chef Tutorial – Chef Nodes**

Nodes can be a cloud based virtual server or a physical server in your own data center, that is managed using central Chef Server. The main component that needs to be present on the Node is an agent that will establish communication with the central Chef Server. This is called Chef Client.

Chef Client performs the following functions:

* 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.

[***Click here to learn how to install Chef Server, Workstation and Node***](https://www.edureka.co/blog/install-chef/)

**Chef Tutorial – 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***](https://www.edureka.co/blog/amazon-aws-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.