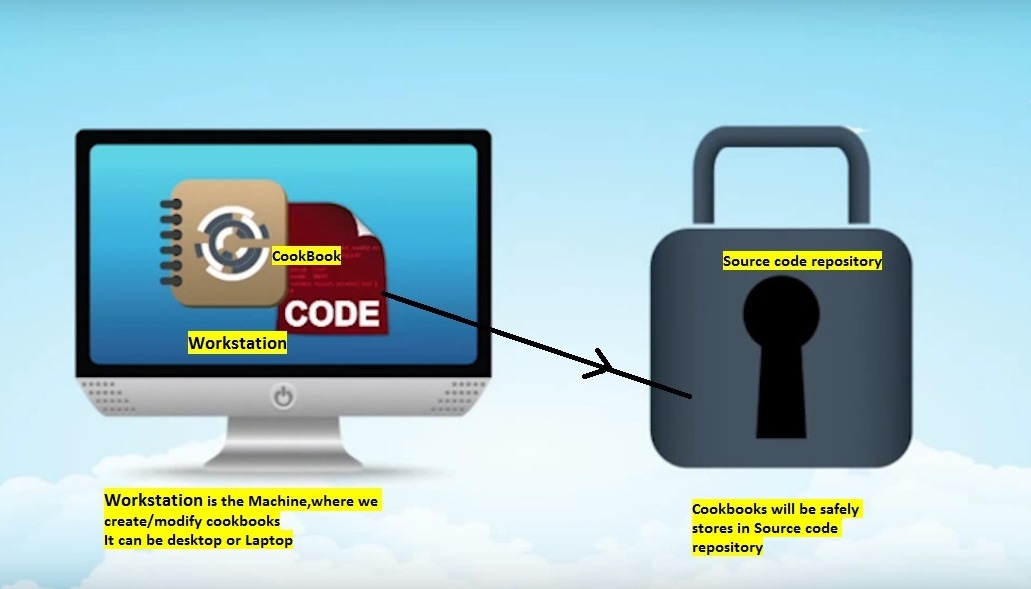
**Chef Workstations**

A workstation is a computer running the Chef Development Kit (ChefDK) that is used to author cookbooks, interact with the Chef server, and interact with nodes.

The workstation is the location from which most users do most of their work, including:

* Developing and testing cookbooks and recipes
* Testing Chef code
* Keeping the chef-repo synchronized with version source control
* Configuring organizational policy, including defining roles and environments, and ensuring that critical data is stored in data bags
* Interacting with nodes, as (or when) required, such as performing a bootstrap operation
* 

The Chef Development Kit tooling encourages integration and unit testing, and defines workflow around cookbook authoring and policy, but it’s important to note that you know best about how your infrastructure should be put together. Therefore, Chef makes as few decisions on its own as possible. When a decision must be made tools uses a reasonable default setting that can be easily changed. While Chef encourages the use of the tooling packaged in the Chef DK, none of these tools should be seen as a requirement or pre-requisite for being successful using Chef.

**Workstation Components and Tools**

Some important tools and components of Chef workstations include:

| **Component** | **Description** |
| --- | --- |
|  | The Chef Development Kit is a package that contains everything that is needed to start using Chef:   * chef-client and ohai * chef command line tool * Testing tools such as Test Kitchen, ChefSpec, Cookstyle, and Foodcritic * Chef provisioning * Everything else needed to author cookbooks and upload them to the Chef server |
|  | ChefDK includes two important command-line tools:   * **Chef:** Use the chef command-line tool to work with items in a chef-repo, which is the primary location in which cookbooks are authored, tested, and maintained, and from which policy is uploaded to the Chef server * **Knife:** Use the knife command-line tool to interact with nodes or work with objects on the Chef server |
|  | The chef-repo is the repository structure in which cookbooks are authored, tested, and maintained:   * Cookbooks contain recipes, attributes, custom resources, libraries, files, templates, tests, and metadata * The chef-repo should be synchronized with a version control system (such as git), and then managed as if it were source code   The directory structure within the chef-repo varies. Some organizations prefer to keep all of their cookbooks in a single chef-repo, while other organizations prefer to use a chef-repo for every cookbook. |
|  | Use Test Kitchen to automatically test cookbook data across any combination of platforms and test suites:   * Defined in a .kitchen.yml file * Uses a driver plugin architecture * Supports cookbook testing across many cloud providers and virtualization technologies * Supports all common testing frameworks that are used by the Ruby community |
|  | Use ChefSpec to simulate the convergence of resources on a node:   * Is an extension of RSpec, a behavior-driven development (BDD) framework for Ruby * Is the fastest way to test resources and recipes |