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

|  |  |
| --- | --- |
|  |  |

knife is a command-line tool that provides an interface between a local chef-repo and the Chef server.

|  |
| --- |
| . knife helps users to manage:   * Nodes * Cookbooks and recipes * Roles, Environments, and Data Bags * Resources within various cloud environments * The installation of the chef-client onto nodes * Searching of indexed data 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.

A knife.rb file is used to specify configuration details for knife

* 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

**About the chef-repo**

The chef-repo is a directory on your workstation that stores:

* Cookbooks (including recipes, attributes, custom resources, libraries, and templates)
* Roles
* Data bags
* Environments

The chef-repo directory should be synchronized with a version control system, such as git. All of the data in the chef-repo should be treated like source code.

knife is used to upload data to the Chef server from the chef-repo directory. Once uploaded, that data is used by the chef-client to manage all of the nodes that are registered with the Chef server and to ensure that the correct cookbooks, environments, roles, and other settings are applied to nodes correctly.

## Directory Structure

The chef-repo contains several directories, each with a README file that describes what it is for and how to use that directory when managing systems.

The sub-directories in the chef-repo are:

| **Directory** | **Description** |
| --- | --- |
| .chef/ | A hidden directory that is used to store validation key files and the knife.rb file. |
| cookbooks/ | Contains cookbooks that have been downloaded from the [https://supermarket.chef.io](https://supermarket.chef.io/) or created locally. |
| data\_bags/ | Stores data bags (and data bag items) in JSON (.json). |
| environments/ | Stores environment in Ruby (.rb) or JSON (.json). |
| roles/ | Stores roles in Ruby (.rb) or JSON (.json). |

### .chef/[¶](https://docs.chef.io/chef_repo.html#chef)

The .chef directory is a hidden directory that is used to store validation key files and the knife.rb file.

### cookbooks/[¶](https://docs.chef.io/chef_repo.html#cookbooks)

The cookbooks/ directory is used to store the cookbooks that are used by the chef-client when configuring the various systems in the organization. This directory contains the cookbooks that are used to configure systems in the infrastructure. Each cookbook can be configured to contain cookbook-specific copyright, email, and license data.

### data\_bags/[¶](https://docs.chef.io/chef_repo.html#data-bags)

The data\_bags/ directory is used to store all of the data bags that exist for an organization. Each sub-directory corresponds to a single data bag on the Chef server and contains a JSON file for each data bag item. If a sub-directory does not exist, then create it using SSL commands. After a data bag item is created, it can then be uploaded to the Chef server.

### environments/[¶](https://docs.chef.io/chef_repo.html#environments)

The environments/ directory is used to store the files that define the environments that are available to the Chef server. The environments files can be Ruby DSL files (.rb) or they can be JSON files (.json). Use knife to install environment files to the Chef server.

### roles/[¶](https://docs.chef.io/chef_repo.html#roles)

The roles/ directory is used to store the files that define the roles that are available to the Chef server. The roles files can be Ruby DSL files (.rb) or they can be JSON files (.json). Use knife to install role files to the Chef server.

## chefignore Files[¶](https://docs.chef.io/chef_repo.html#chefignore-files)

## Create the chef-repo[¶](https://docs.chef.io/chef_repo.html#create-the-chef-repo)

There are two ways to create a chef-repo when using the Chef boilerplate repository as a base:

* Clone the chef-repo from GitHub
* Download the chef-repo as a tar.gz file and place it into local version source control.