**Policy**

Policy **maps business and operational requirements, process, and workflow to settings and objects stored on the Chef server**:

* **Roles define server types, such as “web server” or “database server”**
* **Environments define process, such as “dev”, “staging”, or “production**”
* Certain types of data—**passwords, user account data, and other sensitive items—can be placed in data bags, which are located in a secure sub-area on the Chef server that can only be accessed by nodes that authenticate to the Chef server with the correct SSL certificates**
* The cookbooks (and cookbook versions) in which organization-specific configuration policies are maintained

Some important aspects of policy include:

| **Feature** | **Description** |
| --- | --- |
|  | A **role is a way to define certain patterns and processes that exist across nodes in an organization as belonging to a single job function**. Each role consists of zero (or more) attributes and a run-list. Each node can have zero (or more) roles assigned to it. When a role is run against a node, the configuration details of that node are compared against the attributes of the role, and then the contents of that role’s run-list are applied to the node’s configuration details. When a chef-client runs, it merges its own attributes and run-lists with those contained within each assigned role. |
|  | An **environment is a way to map an organization’s real-life workflow to what can be configured and managed when using Chef server**. Every organization begins with a single environment called the \_default environment, which cannot be modified (or deleted). Additional environments can be created to reflect each organization’s patterns and workflow. **For example, creating production, staging, testing, and development environments. Generally, an environment is also associated with one (or more) cookbook versions.** |
|  | **A cookbook version represents a set of functionality that is different from the cookbook on which it is based**. A version may exist for many reasons, such as ensuring the correct use of a third-party component, updating a bug fix, or adding an improvement. A cookbook version is defined using syntax and operators, may be associated with environments, cookbook metadata, and/or run-lists, and may be frozen (to prevent unwanted updates from being made).  A cookbook version is maintained just like a cookbook, with regard to source control, uploading it to the Chef server, and how the chef-client applies that cookbook when configuring nodes. |
|  | A run-list defines all of the information necessary for Chef to configure a node into the desired state. A run-list is:   * An ordered list of roles and/or recipes that are run in the exact order defined in the run-list; if a recipe appears more than once in the run-list, the chef-client will not run it twice * Always specific to the node on which it runs; nodes may have a run-list that is identical to the run-list used by other nodes * Stored as part of the node object on the Chef server * Maintained using knife, and then uploaded from the workstation to the Chef server, or is maintained using the Chef management console |