



Ohai

Ohai is a tool that is used to detect attributes on a node, and then provide these attributes to the chef-client at the start of every chef-client run. Ohai is required by the chef-client and must be present on a node. (Ohai is installed on a node as part of the chef-client install process.)

The types of attributes Ohai collects include (but are not limited to):

- Platform details
- Network usage
- Memory usage
- CPU data
- Kernel data
- Host names
- Fully qualified domain names
- Other configuration details

Attributes that are collected by Ohai are automatic attributes, in that these attributes are used by the chef-client to ensure that these attributes remain unchanged after the chef-client is done configuring the node.

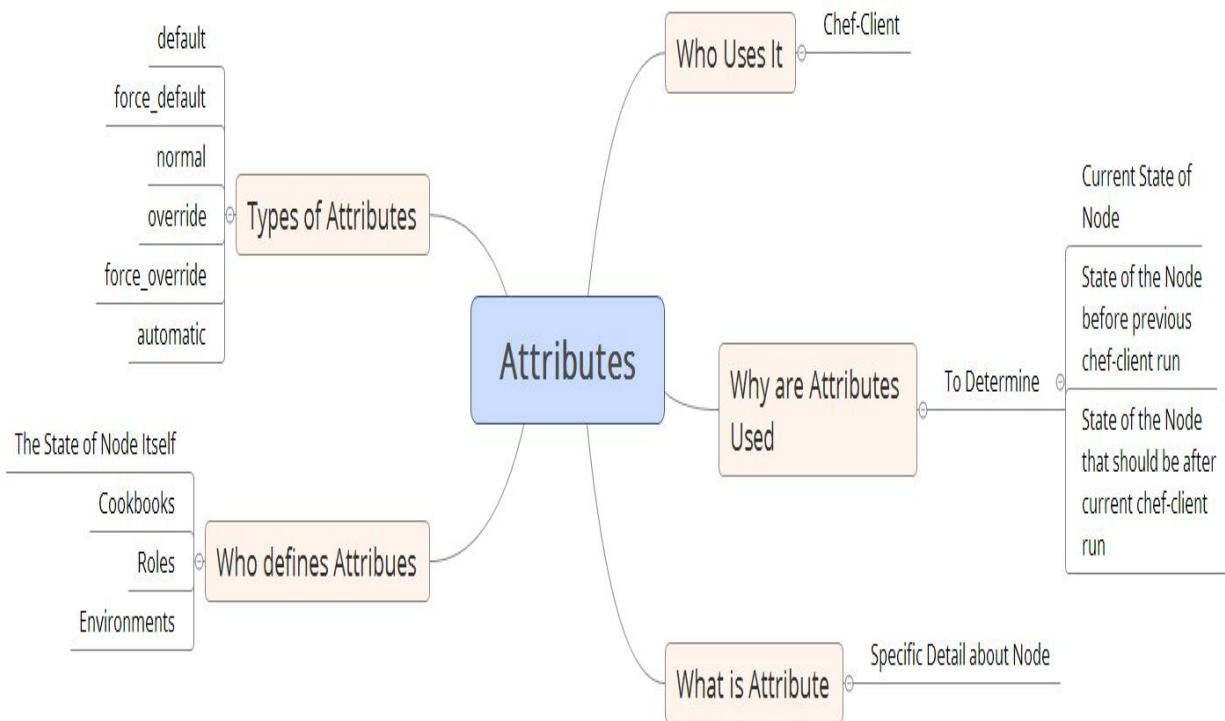
Attributes

An attribute is a specific detail about a node. Attributes are used by the chef-client to understand:

- The current state of the node
- What the state of the node was at the end of the previous chef-client run
- What the state of the node should be at the end of the current chef-client run

Attributes are defined by:

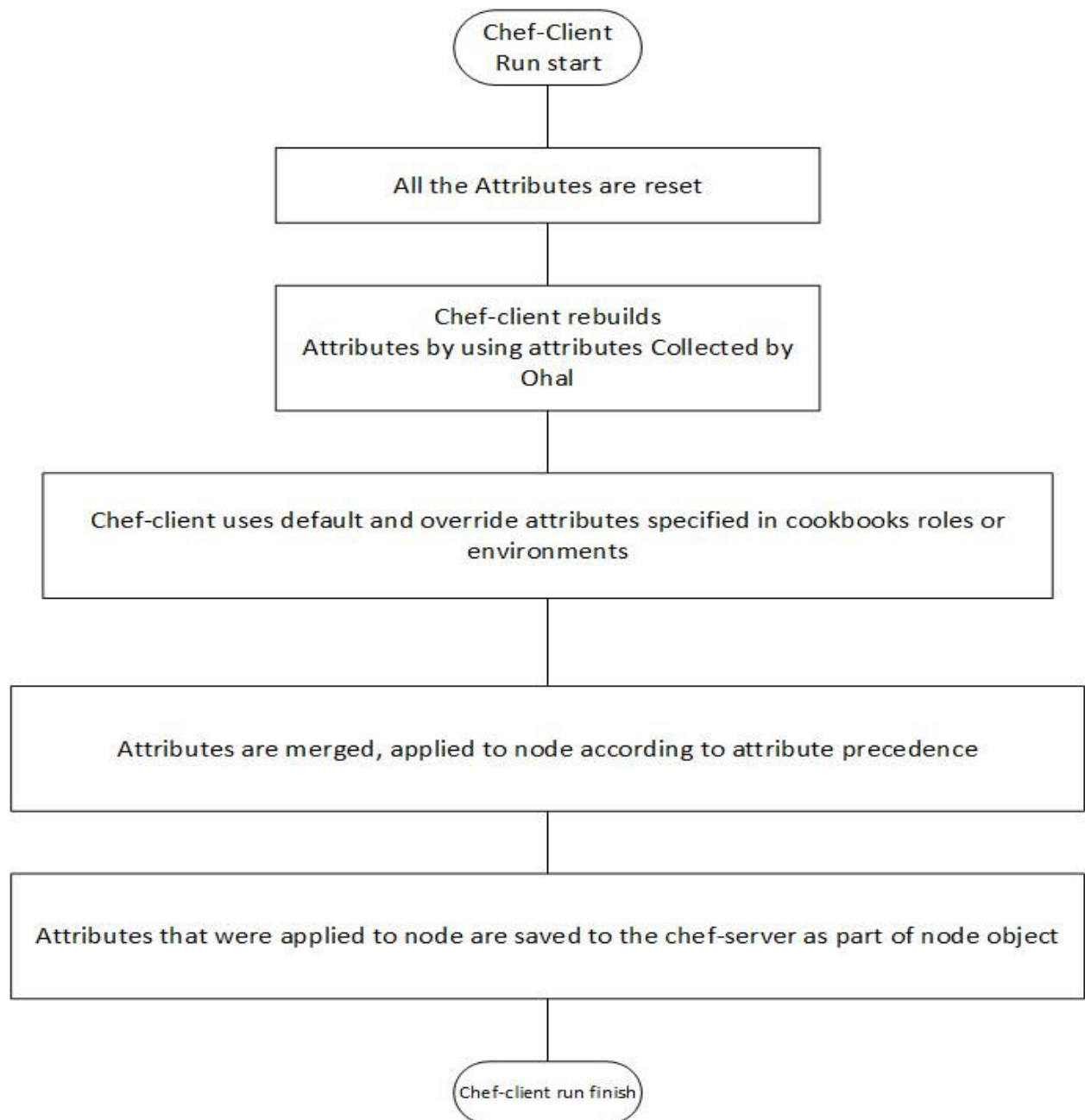
- The state of the node itself
- Cookbooks (in attribute files and/or recipes)
- Roles
- Environments



During every chef-client run, the chef-client builds the attribute list using:

- Data about the node collected by Ohai
- The node object that was saved to the Chef server at the end of the previous chef-client run
- The rebuilt node object from the current chef-client run, after it is updated for changes to cookbooks (attribute files and/or recipes), roles, and/or environments, and updated for any changes to the state of the node itself

After the node object is rebuilt, all of the attributes are compared, and then the node is updated based on attribute precedence. At the end of every chef-client run, the node object that defines the current state of the node is uploaded to the Chef server so that it can be indexed for search



Attribute Types

default

Automatically reset at every chef-client run

Lowest Precedence

recommended attribute type

force_default

Use the force_default to ensure that attribute defined in cookbook takes precedence over default defined in roles or environment

normal

Normal attributes are never reset

Normal attribute persists in node object

Higher precedence than default attribute

override

An override attribute is automatically reset at the start of every chef-client run

higher attribute precedence than default, force_default, and normal attributes

An override attribute is most often specified in a recipe, but can be specified in an attribute file, for a role, and/or for an environment.

A cookbook should be authored so that it uses override attributes only when required.

force_override

Use the force_override attribute to ensure that an attribute defined in a cookbook (by an attribute file or by a recipe) takes precedence over an override attribute set by a role or an environment.

automatic

An automatic attribute contains data that is identified by Ohai at the beginning of every chef-client run

An automatic attribute cannot be modified and always has the highest attribute precedence.

Attribute Precedence

	Attribute Files	Node / Recipe	Environment	Role
default	1	2	3	4
force_default	5	6		
normal	7	8		
override	9	10	12	11
force_override	13	14		
automatic			15	

Node Object

Node object is made up of the run lists which define what recipes to run during a chef-client as well as the attributes that define information about the node

Attributes are built during the chef-client run process:

- Data about the node is collected by Ohai
- The node object previously saved during the last chef-client run
- The rebuilt node object from the current chef-client run

Once the node object is rebuilt all attributes are compared and then updated based on attribute precedence

At the end of every chef-client run the node object that defines the current state of the node is uploaded to the chef server to be searched.

Search:

Chef search allows queries to be made for any type of data that is indexed by the chef server. Search queries the chef server for stored information.

Can Search:

- Data bags
- Environments
- Roles
- Nodes

Search: Methods of search

Search with Knife:

Syntax: `knife search node "key:pattern" -a (attribute)`

- Search nested attributes
 - `"memory_total:*" -a memory.total`
- Can use basic "wild cards" in the pattern `"ipaddress:192.168.*"`
- Can search based off ranges `"ipaddress:[192.168.* TO 192.172.*]"`

Search inside of recipes

Example:

Use Ruby recipe to search all nodes that are running an outdated package

Data Bags

A data bag is a global variable that is stored in JSON data and accessible from the chef server. A data bag is indexed for searching and can be loaded by a recipe or accessed during search.

Types of data stored in a data bag:

- ✓ Users to be added to a system
- ✓ Admins to be added to sudo
- ✓ API/DB Credentials (More secure and better than environment attributes for credentials)
- ✓ Much more