**Nodes**

A node is any machine—physical, virtual, cloud, network device, etc.—that is under management by Chef.



**Node Types**

The types of nodes that can be managed by Chef include, but are not limited to, the following:

| **Node Type** | **Description** |
| --- | --- |
|  | A **physical node** is typically a server or a virtual machine, but it can be any active device attached to a network that is capable of sending, receiving, and forwarding information over a communications channel. In other words, a physical node is any active device attached to a network that can run a chef-client and also allow that chef-client to communicate with a Chef server. |
|  | A **cloud-based** node is hosted in an external cloud-based service, such as Amazon Web Services (AWS), OpenStack, Rackspace, Google Compute Engine, or Microsoft Azure. Plugins are available for knife that provide support for external cloud-based services. knife can use these plugins to create instances on cloud-based services. Once created, the chef-client can be used to deploy, configure, and maintain those instances. |
|  | A **virtual node is a machine** that runs only as a software implementation, but otherwise behaves much like a physical machine. |
|  | A **network node is any networking device—a switch, a router**—that is being managed by a chef-client, such as networking devices by Juniper Networks, Arista, Cisco, and F5. Use Chef to automate common network configurations, such as physical and logical Ethernet link properties and VLANs, on these devices. |
|  | Containers are an approach to virtualization that allows a single operating system to host many working configurations, where each working configuration—a container—is assigned a single responsibility that is isolated from all other responsibilities. Containers are popular as a way to manage distributed and scalable applications and services. |

**Chef on Nodes**

The key components of nodes that are under management by Chef include:

| **Component** | **Description** |
| --- | --- |
|  | A chef-client is an agent that runs locally on every node that is under management by Chef. When a chef-client is run, it will perform all of the steps that are required to bring the node into the expected state, including:   * Registering and authenticating the node with the Chef server * Building the node object * Synchronizing cookbooks * Compiling the resource collection by loading each of the required cookbooks, including recipes, attributes, and all other dependencies * Taking the appropriate and required actions to configure the node * Looking for exceptions and notifications, handling each as required   RSA public key-pairs are used to authenticate the chef-client with the Chef server every time a chef-client needs access to data that is stored on the Chef server. This prevents any node from accessing data that it shouldn’t and it ensures that only nodes that are properly registered with the Chef server can be managed. |
|  | 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 * Virtualization data * Cloud provider metadata * Other configuration details   Attributes that are collected by Ohai are automatic level 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. |