

Puppet Classes

"Resources are the fundamental unit for modeling system configurations. They're like building blocks. Think 'Lego'. Each resource describes some aspect of a system, like a service that must be running or a package that must be installed. The block of code that describes a resource is called a resource declaration. Resources can be combined together to represent the desired configuration of your system. And order doesn't matter."

"Classes define a collection of resources that are managed together as a single unit. Stated another way, package, file, and service are individual Puppet resources bundled together to define a single idea or class. Class definitions are contained in manifests. A class contains all of its resources. This means, any relationship formed with the class as a whole will be extended to every resource in the class."

"The general form of a class definition is: the class keyword; the name of the class; and while the following is outside the scope of this course, it is important that you are aware of other information that you might find in a class definition."

"After the class name, before the opening curly brace, you might find an optional set of parameters which consist of: an opening parenthesis; a comma separated list of parameters; , each of which consists of a new variable name including the dollar sign prefix, an optional equal sign, and a default value; an optional trailing comma might be found after the last parameter, and a closing parenthesis. Also, you might find the `inherit` keyword, followed by a single class name. After these optional parameters, there is an opening curly brace followed by a block of Puppet code, which generally contains at least one resource declaration. These are the resources you're defining that will be managed as a single unit. And then you'll find a closing curly brace."

"Let's look at two important aspects of the behavior of a class. First is the definition. To define a class is to specify the contents and behavior of a class. Defining a class doesn't automatically include it in a configuration. It simply makes it available to be declared. Declaring a class adds a resource to the catalog, and tells Puppet to manage that resource's state. When Puppet applies the compiled catalog, it will read the actual state of the resource on the target system, compare the actual state to the desired state, and if necessary, change the system to enforce the desired state."

"After defining a class, you must declare the class in order to use it in your systems configuration. Declaring a class directs Puppet to include or instantiate a given class. To declare a class, use the **include** keyword or the syntax that you see on your screen, where **class** followed by an open curly brace, the name of the class in double quotes, followed by a colon, closing curly brace, also declare a class."

"If a more complex deployment is needed, reusing existing classes saves effort and reduces

error. Classes are reusable and singleton. Classes are reusable in that they can be used across multiple nodes. However, classes must be unique and once declared, classes can not be redeclared."

"Multiple classes can be declared together to represent a role. This is a node definition, which represents the agent machine and the classes that compose its Puppet configuration. More specifically, we're building a web application from Puppet classes on `site.example.com.`, and declaring that the `ssh`, `apache`, `mysql`, and `web-app` classes be included in Puppet's management of this node."

"In summary, the core of the Puppet language is the resource declaration. A resource declaration describes a desired state for one resource. Resources are then combined into single units called classes. Once classes are defined, they must be declared in order for Puppet to manage the associated resources. And as you will see in later courses, a manifest will hold these class definitions as well as other logic for Puppet to use."

"To complete this course, complete the short quiz and work through any exercises that are found immediately below this video."