

Course Introduction

Introduction to Automation at Scale

Introduction to Puppet

✔ Video: What is Puppet? 3 min

✔ Video: Puppet Resources 3 min

✔ Video: Puppet Classes 2 min

✔ Reading: Puppet Resources 10 min

📋 Practice Quiz: Practice Quiz: Introduction to Puppet 5 questions

The Building Blocks of Configuration Management

Module 1 Review

✔ Congratulations! You passed!

Grade received 100%. To pass 80% or higher

Practice Quiz • 10 min

Go to next item

1. A Puppet agent inspects /etc/conf.d, determines the OS to be Gentoo Linux, then activates the Portage package manager. What is the provider in this scenario?

1 / 1 point

☐ /etc/conf.d

☒ Portage

☐ Gentoo Linux

☐ The Puppet agent

✔ Correct

You got it! The Portage package manager used by Gentoo Linux is the provider called by the Puppet agent.

👍 Like

👎 Dislike

📄 Report an issue

✔ Receive grade

To Pass 80% or higher

Your grade

100%

View Feedback

We keep your highest score

Try again

2. Which of the following examples show proper Puppet syntax?

1 / 1 point

☒

```
1 class AutoConfig {
2   package { 'Executable':
3     ensure => latest,
4   }
5   file { 'executable.cfg':
6     source => 'puppet:///modules/executable/Autoconfig/executable.cfg'
7     replace => true,
8   }
9   service { 'executable.exe':
10    enable => true,
11    ensure => running,
12  }
13 }
14
```

☐

```
1 class AutoConfig :
2   package 'Executable':
3     ensure => latest,
4
5   file 'executable.cfg':
6     source => 'puppet:///modules/executable/Autoconfig/executable.cfg'
7     replace => true,
8
9   service 'executable.exe':
10    enable => true,
11    ensure => running,
12
```

☐

```
1 class AutoConfig {
2   package { 'Executable':
3     ensure == latest,
4   }
5   file { 'executable.cfg':
6     source == 'puppet:///modules/executable/Autoconfig/executable.cfg'
7     replace == yes,
8   }
9   service { 'executable.exe':
10    enable == yes,
11    ensure == true,
12  }
13 }
14
```

☐

```
1 class AutoConfig {
2   package { 'Executable':
3     assure=> latest,
4   }
5   file { 'executable.cfg':
6     origin=> 'puppet:///modules/executable/Autoconfig/executable.cfg'
7     substitute=> true,
8   }
9   program{ 'executable.exe':
10    activate => true,
11    assure => running,
12  }
13 }
14
```

✔ Correct

Excellent! The AutoConfig class has all its resources grouped together using proper Puppet syntax.

3. What is the benefit of grouping resources into classes when using Puppet?

1 / 1 point

☐ Providers can be specified

☒ Configuration management is simplified

☐ The title is changeable

☐ Packages are not required

✔ Correct

Right on! Grouping a collection of related resources into a single class simplifies configuration management by, for one example, allowing us to apply a single class to each host rather than having to specify every resource for each host separately and possibly missing some.

4. What defines which provider will be used for a particular resource?

1 / 1 point

☒ Puppet assigns providers based on the resource type and data collected from the system.

☐ A menu allows you to choose providers on a case-by-case basis.

☐ The user is required to define providers in a config file.

☐ Puppet uses an internet database to decide which provider to use.

✔ Correct

Awesome! Puppet assigns providers according to predefined rules for the resource type and data collected from the system such as the family of the underlying operating system.

5. In Puppet's file resource type, which attribute overwrites content that already exists?

1 / 1 point

☐ Purge

☐ Overwrite

☒ Replace

☐ Save

✔ Correct

Oh yeah! Puppet has many useful attributes. "Replace" set to True tells Puppet to replace files or symlinks that already exist on the local system but whose content doesn't match what the source or content attribute specifies.