

✓ ¡Felicitaciones! ¡Aprobaste!

Calificación recibida 100 % Para Aprobar 80 % o más

Ir al siguiente elemento

!

Estás viendo una versión traducida automáticamente de esta evaluación

Puedes volver a ver el contenido en su idioma original si lo prefieres. No perderás el progreso que hayas conseguido si cambias el idioma.

Mostrar la versión en Inglés

×

1.

Un agente de Puppet inspecciona /etc/conf.d, determina que el sistema operativo es Gentoo Linux y activa el gestor de paquetes Portage. ¿Cuál es el proveedor en este escenario?

1 / 1 punto

/etc/conf.d

Portage

Gentoo Linux

El agente Puppet

✓ Correcto

¡Lo ha conseguido! El gestor de paquetes Portage utilizado por Gentoo Linux es el proveedor llamado por el agente Puppet.

2.

¿Cuál de los siguientes ejemplos muestra la sintaxis correcta de Puppet?

1 / 1 punto

```
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
```

✓ Correcto

¡Excelente! La clase AutoConfig tiene todos sus recursos agrupados utilizando la sintaxis adecuada de Puppet.

3.

¿Cuál es la ventaja de agrupar los recursos en clases cuando se utiliza Puppet?

1 / 1 punto

Se pueden especificar los proveedores

La gestión de la configuración se simplifica

El título se puede cambiar

Los paquetes no son necesarios

✓ Correcto

Muy bien Agrupar una colección de recursos relacionados en una única clase simplifica la gestión de la configuración, ya que, por ejemplo, nos permite aplicar una única clase a cada host en lugar de tener que especificar cada recurso para cada host por separado y posiblemente omitir alguno.

4.

¿Qué define qué proveedor se utilizará para un recurso concreto?

1 / 1 punto

- Puppet asigna los proveedores en función del tipo de recurso y de los datos recogidos del sistema.
- Un menú le permite elegir a los proveedores en función de cada caso.
- El usuario debe definir los proveedores en un archivo de configuración.
- Puppet utiliza una base de datos de Internet para decidir qué proveedor utilizar.

✓ Correcto

Impresionante Puppet asigna los proveedores según reglas predefinidas para el tipo de recurso y los datos recogidos del sistema, como la familia del sistema operativo subyacente.

5.

En el tipo de recurso archivo de Puppet, ¿qué atributo sobrescribe el contenido que ya existe?

1 / 1 punto

- Purga
- Sobrescribir
- Sustituya
- Guarde

✓ Correcto

¡Ah, sí! Puppet tiene muchos atributos útiles. "Reemplazar" en True le dice a Puppet que reemplace los archivos o symlinks que ya existen en el sistema local pero cuyo contenido no coincide con lo que especifica el atributo source o content.